PRELIMINARY STEPS IN OPTIMIZING UNIVERSITY COMPUTER PERFORMANCE USING HARDWARE AND SOFTWARE MONITORS

Robert Richard Hanke

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PRELIMINARY STEPS IN OPTIMIZING
UNIVERSITY COMPUTER PERFORMANCE
USING HARDWARE AND SOFTWARE MONITORS

by

Robert Richard Hanke

Thesis Advisor: G. H. Syms

December 1971

Approved for public release; distribution unlimited.

Preliminary Steps in Optimizing University Computer Performance

Using Hardware and Software Monitors

by

Robert Richard Hanke
Lieutenant, United States Navy
B.A., Northwestern University, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 1971

ABSTRACT

Too much money is being spent on new computer systems without any idea of what the new systems can do. The large expenditures for computer hardware necessitate obtaining the maximum performance for every dollar spent, in order for the computer system to be cost effective.

This research effort explores the process of selecting, implementing, and using a hardware monitor to measure the performance of a university computer system. Information about the work being performed by the computer system was obtained without the use of a special software monitor, instead the System Management Facilities data files were read to obtain job stream data.

System performance profiles were obtained to indicate the utilization of system resources. Recommendations are made to isolate the cause of the central processing unit waiting for the selector channel to complete input/output operations, which would improve the overall performance of the computer system.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

From the available literature on the evaluation of a computer system, it appears that everyone agrees that technology has pushed hardware development far beyond the limits of current evaluation techniques. Before the complete relationships between system components can be understood and system elements can be rearranged for improved performance, the analyst must know what the system is doing, what resources it is using, and why it is using them. Many of these questions can be answered by using hardware and software monitors to measure the operation of a computer system. For example, C. Dudley Warner [Ref., 1] details the needs for, and advantages of system evaluation using a hardware monitor.

Donald R. Deese [Ref. 2] describes as a must, the determination of the user environment when measuring the performance of a computer system. Thus, it was necessary to develop a program to record the workload being placed upon the computer system during periods of measurement. This program served as the software monitor for the experiments performed.

A number of sources in the available literature suggest a system performance measurement experiment that should be made. This experiment essentially measures the average utilization of components and sets of components. The systems analyst uses this information to

balance the computer system and eliminate bottlenecks which are reducing system efficiency. A guide is provided to aid the systems analyst in interpreting the results of a system performance profile.

This thesis presents results of several system performance profiles that were performed on the IBM 360 Model 67 at the Naval Postgraduate School. Recommendations are presented to improve the system performance.

II. OBJECTIVES

The almost complete lack of information regarding the performance of a modern computing system and how to improve that performance was the prime motivation for undertaking the evaluation of the IBM 360 at this institution. Again and again the students heard that the load on the school's computer was an unknown factor and performance could not be measured without data on this workload.

During the summer of 1971 this institution acquired a hardware monitor to measure the performance of its computer system. This provided the equipment to begin this thesis project and a glimpse at the problems encountered when implementing and using a hardware monitor.

Before steps can be taken to optimize the performance of a computer system using a hardware monitor, a specific hardware monitor must be selected. Donald R. Deese [Ref. 2] stated that best results are obtained from any measurement device when the objectives of that measurement are clearly understood. Knowing what the hardware monitor will measure and how it will be used, are the first steps towards improving performance. Like all computer system hardware there are numerous sources for purchasing, leasing, or obtaining service of a hardware monitor. Hardware monitors come in all sizes, shapes, and price ranges. Just choosing a hardware monitor is a formidable task. This thesis provides a guide to selection of a specific hardware monitor.

Implementation of a hardware monitor can be filled with problems. Determining what staff is required to use this new equipment is most critical to the successful measurement of computer performance. Mark J. McGrew, Executive Vice President, Allied Computer Technology, in correspondence with the author, stated that the ideal candidate for the use of measurement equipment is a data processing professional with eight to twelve years experience in system programming and application design. A secondary objective of this paper is to explore the complex problems encountered when implementing a hardware monitor.

What measurements to make with a hardware monitor is surely the most crucial objective of this paper. Specific experiments are described and sources of further information on what to measure are discussed.

III. SELECTING A HARDWARE MONITOR

This section is concerned with the selection of a hardware monitor from the numerous devices that are available today. Mr. L. E. Hart [Ref. 3] provides an alphabetical listing of the various companies involved in computer evaluation with which he has had experience. Software and hardware devices for computer evaluation are listed in this guide. It is an excellent starting point for the manager who is not aware of the companies producing hardware monitors. Letters were sent to the firms listed by Mr. Hart and descriptions of the products of the firms who responded are included in this section.

Before discussing the selection of a monitor, the major components of a hardware monitor will be discussed. A hardware monitor is composed of four major components. The probes or sensors of the hardware monitor unit attach to the host computer system to sense the occurrence of signals in the host without degrading the performance of that host system. The logic panel of the monitor unit then combines ~~the~~ and routes probe inputs logically according to the user-connected patch board. The accumulators serve as the memory for the hardware unit during each recording interval. The accumulators are used to store such values as the percent utilization, occurrences of an event during the interval, or total number of occurrences of an event. The last component, a recording unit, serves as the permanent memory of the monitor unit. It is

usually a magnetic tape device, but may be only a paper tape printer on less sophisticated devices.

Some type of data reduction program is normally applied to the measurement results stored on the recording unit to produce a series of analysis and summary graphs and tables of the host computer system's operations. The analyst now has a graphic presentation of exactly what the host system is doing with its available resources. Figure 1 is a typical hardware monitor system in block form. The rest of this section describes how to select the components of a hardware monitor.

The selection of a particular model hardware monitor is much like the selection of any piece of computer hardware. Mr. A. J. Bonner [Ref. 4] points out the dangers inherent in the wide choice of compatible hardware units. One can easily "tailor-make" a very inefficient complex system. Spending more is not the key to success with any computer hardware acquisition, and hardware monitors follow this trend. Components must be selected for the monitor which will meet the needs of the individual data center. Output devices for hardware monitors range from simple paper tapes to high speed magnetic tape units. Some monitors have software packages which take the raw data and prepare finished reports for management use. One manufacturer even offers a monitor package which produces simulated hardware monitor data from a simulated computer system. The computer center manager can thus explore configuration changes with his hardware monitor. This particular unit uses the job stream of the computer center as a basis for its simulation.

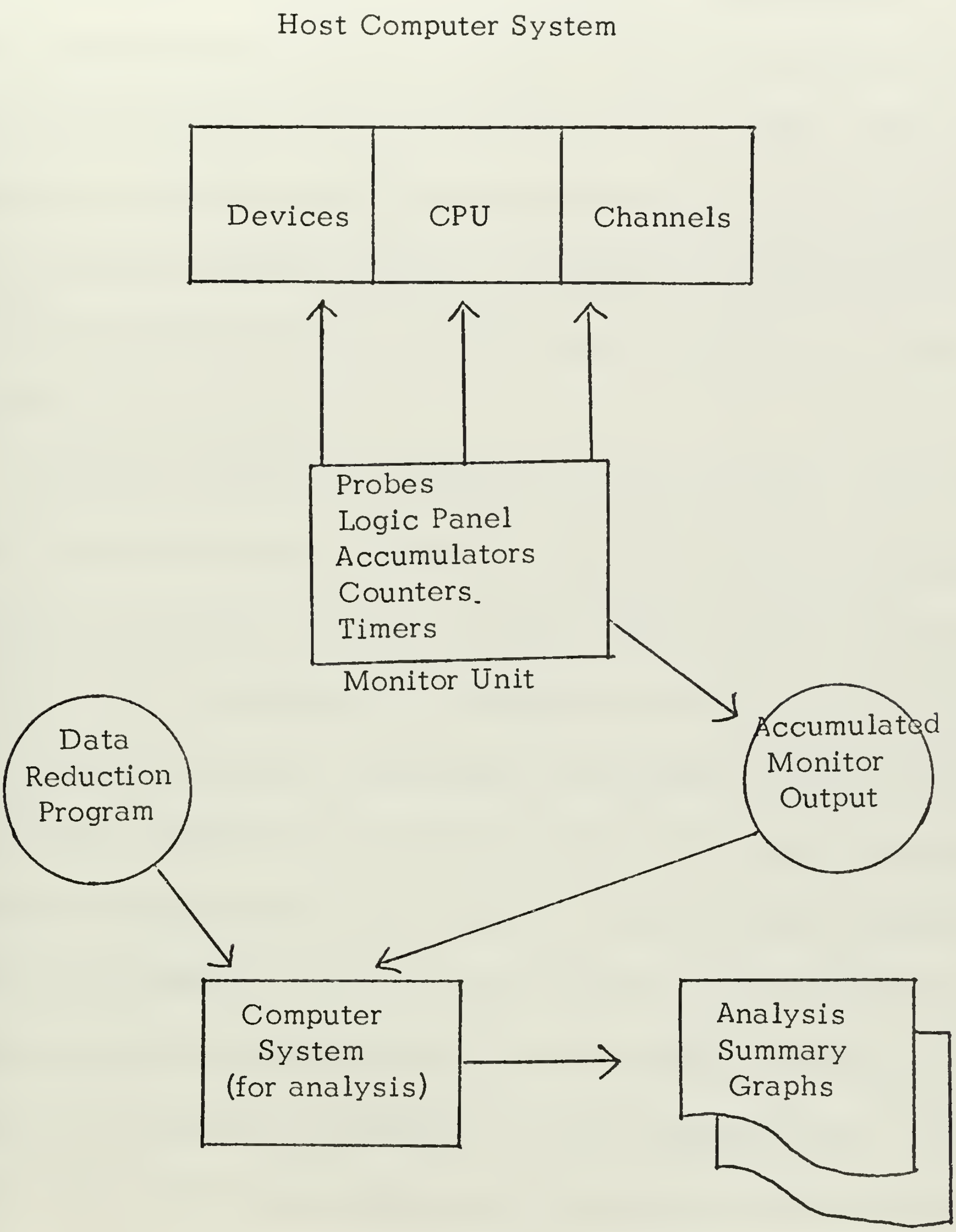


Figure 1. Hardware Monitor System

Mr. Donald R. Deese [Ref. 2] states that without exception it has been his experience that the best results can be obtained in optimizing the performance of a computer system by use of a hardware monitor when specific problem areas have been defined. The monitor is then used to find the cause of these problems. The worst results are obtained when the user monitors his system with no goals in mind. The determination of evaluation goals is the key to success in any measurement of the computer system performance. Spending fifty thousand dollars for the best hardware monitor available will not provide the solution to any performance problem.

Hardware monitors are available for sale, for lease, and service bureau firms even offer to operate your equipment or bring in their own monitoring equipment. Mr. C. D. Warner [Ref. 1] believes that the large mainframe manufacturers of computer hardware will not enter the hardware monitor market. The different performance obtained by users of the same computer system is likened to the wide ranges in gasoline mileage reported by owners of similar cars. Specifically, the environment at each computer center is a major factor in determining the performance of any computer system. Mr. Warner also emphasizes the problem of identifying the problems in your computer system. A list of needs and intended applications should be used as a basis for evaluating the products of various manufacturers. Unlike the mainframe hardware which is mostly rented, hardware monitors are today mostly purchased. This is undoubtedly because of their smaller cost and the smaller assets of the manufacturers of hardware monitors.

Besides the vast differences in the output equipment available for hardware monitors, there is a vast difference in capabilities of the basic counter and integrating units. All monitors consist of a set of probes, which passively monitor the computer system signals, and a logic unit, which takes the signals received and logically combines them prior to producing results. The physical size of the monitor devices varies from huge one thousand pound monsters, which are barely portable units, to small units about the size of a portable record player. The larger units include an internal magnetic tape drive and printer as well as multiple programable logic plugboards. Modular construction of some hardware monitors offers the manager the easiest choice. A small and less expensive basic unit can be purchased. Multiples of the same units along with higher speed and greater capacity data reduction units can be added at a later date as the knowledge and needs increase. Many manufacturers offer units which exchange data with similar units. Thus, the information which one logic panel calculates from several probe points can be passed to a similar logic panel for further processing.

Careful consideration should be given to the size of the logic plugboards of each hardware monitor. A device which can accept the inputs of thirty probes but contains a limited number of FANOUTS, AND gates, OR gates, and INVERTERS will be of limited value. Another difficulty can arise if there are frequent changes made in the wiring of logic boards because mistakes occur in rewiring and experiments are delayed.

The final configuration of a logic board for an experiment should be recorded for future checking as well as for repetition of the same experiment in the future.

The following four subsections describe the hardware monitors of four manufacturers. They are Boole & Babbage, Compress, Computer Synectics, and Allied Computer Technology. These four manufacturers supplied the information presented in reply to correspondence inquiring about their system. The information from each firm is necessarily sales department flavored, since the manuals and pamphlets were designed for prospective customers.

A. MEASUREMENT ENGINE

Boole & Babbage produce a computer hardware measurement device which they call the Measurement Engine. This monitor features a compact size and small initial cost for the basic event monitor and printer units. A magnetic tape unit is also available as an option. The logic capability of the event monitor's plugboard may be extended by an optional larger plugboard. Modularity of design enables multiple event monitors to share signals from the measurement probes. This institution has two event monitors and one printer. Section V details the features of the Measurement Engine.

B. DYNAPROBE

Compress produces a line of hardware measurement devices which they refer to as Dynaprobe. Dynaprobe-7700 is a modular line of

computer performance monitors. The minimum number of counters is six, which are six digits wide, and can be expanded up to a maximum of eighteen electromechanical or electronic counters which are 12 digits wide. Extended FANOUT, AND, OR, and NOR logic is provided by the D-7719 Supplementary Probe/Logic Unit. This unit also provided additional probe receivers and additional hexadecimal decoding capability. The D-7712 Output Printer provides the capability for unattended operation of the D-7700. Printer formats include both columnar and (graphic) histogram reporting of monitor readings. Readings may be obtained at preset intervals or asynchronously, under host event control. The D-7720 Comparator extends the capability of the Compress computer performance measurement systems. It compares multi-bit data with predetermined values, passing the results of the comparison to the Dynaprobe. No unit of the D-7700 series is larger than 10" x 19" x 10" and the heaviest unit weighs 49 pounds. Probes are capable of sensing pulses from ±0.25 to ±60 volts with a 30-nanosecond sensitivity. Price for the D-7700 ranges from $5000 for the six counter unit to $10000 for the eighteen counter unit. The other D series units are extra with the exception of the probes which are provided with every unit.

The D-7800 series is a newer member of the Compress line. Coupled with DYNAPAR data reduction software and the largest library of probe points available, the D-7800 represents a computer management tool of the first rank. The Dynaprobe-7800 is composed of the D-7816 Monitor and Magnetic Tape Buffer and the D-7817 Magnetic Tape Unit.

The D-7816 Monitor provides sixteen ten-digit counters to accumulate time or count readings of up to thirty-two probed system functions. The contents of the sixteen counters are written to the D-7817 Tape Drive under manual or logic control along with the contents of the D-7816 Real Time Clock and the settable identification register. Readings produced on the IBM-compatible D-7817 Tape are input to the selected DYNAPAR program to structure the accumulated data into systems performance reports which facilitate analysis by the user. D-7800 is fully buffered. Counts are not lost while tape records are written. The Real Time Clock has precision to seven decimal digits and resolution to the nearest 0.1 second. A ten-digit display register shows the contents of any data register, the Clock or the ID Register. DYNAMAP Program Profile is a feature which provides program activity indicating core areas measured and the percentage of time that the program spent in each area. The D-7816 and the D-7817 weigh 145 pounds and are priced at $25000.

The D-7900 series is the top of the Compress Monitor Line. The D-7817 Magnetic Tape Unit is combined with the D-7916 Monitor/Tape Buffer to form the basic performance monitor system. The major difference in the D-7800 and the D-7900 monitor units appears to be the addition of twelve variable speed counters on the D-7916 Monitor. Twelve single bit variable speed counters expand the capacity of the D-7916 counters from sixteen to twenty-eight. Each variable speed counter may be scaled with D-7916 Logic Panel Scalers (the size of the Scaler constructed determines the accumulating rate-up to 20 Mhz). The weight

of the monitor and tape unit is the same as the D-7800 series and the price is $27000 for the D-7916 Monitor/Tape Buffer and the D-7817 Magnetic Tape Unit.

The D-8000 Programmed Monitor is the ultimate in hardware monitoring. It is a 16-bit programable mini-computer. The D-7900 series unit is the basis for measurements, but now control is performed by the D-8011 Data Handler. The D-7916 Monitors are multiplexed through the D-7818 Multiplexor along with the D-8011 Data Handler. This combines the measurement data on one tape. The analyst can measure hardware and software activity and event sequencing. Mr. D. R. Deese of Compress described the D-8000 as an "add-on" to the D-7900. Its price is $23000, which makes the basic monitor system cost over $50000.

C. SYSTEM UTILIZATION MONITOR

System Utilization Monitor (SUM) is the hardware monitor line of Computer Synectics. SUM connects directly to all major computer manufacturers' equipment without special interfaces or hardware modifications. Monitoring points in the user's system are defined by Computer Synectics. Emphasis is placed on the ease with which reports are made by the accompanying software package. An exclusive feature of SUM is a sensor simulator unit which enables the user to checkout patch panel logic before making a measurement run on the computer.

Sixteen hardware counters are standard with up to thirty-two counters being available as an option. In addition to the hardware counters, nineteen software counters are also available. Time is kept

by a software clock if the hardware clock option is not installed in the user's unit. Twenty sensors are standard with forty sensors available as an option. Up to twenty data comparators may be added to measure directly all types of parallel-word data for memory or storage mapping.

A software clock is standard, but an optional hardware clock is available with a key interlock. This allows direct simplified correlation of real time operation logs of the host computer.

The most unique feature of the SUM system is the Configuration Simulator. It is one of the major features incorporated into Computer Synectics' Advanced SUM Analysis Program (A-SUMDAP). A-SUMDAP is supplied with the SUM unit, thus allowing the user to combine hardware monitoring techniques for data collection with software data reduction analysis programs to provide total performance reporting. Using the initial SUM measurements of the host computer system as a base, the Configuration Simulator enables the user to apply simulation techniques to areas of questionable performance. The Configuration Simulator produces the same management reports that the SUM unit normally produces, but the performance is based on the calculation of the configuration's effect on system functions. Faster and slower CPU's, faster and slower tape and disk storage, and reconfiguration of devices on selector channels are only a few of the possible applications for this feature. The price of SUM varies from $21000 to $50000 depending upon options.

D. COMPUTER PERFORMANCE MONITOR II

Computer Performance Monitor II (CPM-II) is the hardware product of Allied Computer Technology. The CPM-II is equipped with twenty measurement probes and sixteen counters to measure the activity of system functions. Each counter is ten decimal digits wide. Counters can measure the length of time a function is active or count the number of times a function occurred.

A 450 hub removable control panel provides the connection between the probes and the counters. It provides the ability to logically combine functions. An internal hardware clock provides 24 hour measurement of time down to 100 microseconds. The SUM unit is capable of producing a tape report every 100 milliseconds. A visual display of any of the ten digit decimal counters is instantly available for intermediate readings and for diagnostic check-out.

A nine track, 800 bpi, 20kc, synchronous tape recorder with a 1200 foot reel capacity is provided with each CPM-II. A Comparator Feature provides as an option 24 bit comparison at a 200 nanosecond rate. It may be used to facilitate subroutine timing, memory utilization measurements, and other previously opaque measurements. Edge transition triggers provide the ability to recognize both signal transitions. Larger hub control panels and additional counters are available along with other options which enable the user to tailor the CPM-II to his needs. This system is 48" x 39" x 49" and weighs 450 pounds. It is not as portable as the Dynaprobe, Boole & Babbage, or the SUM product lines.

CPM-II is priced at $43000 for an average unit depending on options selected.

Figure 2 is a summary of the four systems that have just been explored in detail. These four systems are not necessarily the best ones available. Several other excellent monitor systems are available. Clasco Systems produces X-RAY which is comparable in size to the SUM system and appears to offer about the same options. X-RAY does offer up to 96 sensors and 32 counters which are 32 digits wide. It is a mini-computer like the Dynaprobe-8000 series and surely has a price tage in the $50000 range.

IBM produced the first widely known hardware unit, the Basic Counting Unit. For a short time it was available at no charge to IBM customers to make basic performance measurements. IBM people in the field were not well trained in its use. Basic measurements designed to search for a balanced system were made, but it was not always obvious what the results meant and what should be done to get a balanced system. A charge is now made for use of an upgraded BCU.

Product Name	Counters	Digits/ Counter	Tape Unit Available	Comparator Available	Max Probes	Visual Display	Min Pulse Width (Nanosecs)	Size (inches)	Cost
Measurement Engine	6	4	Yes	No	16	Yes	50	17x12x4	$7000
Dynaprobe-7700	6, 12, 18	6, 12	Yes	Yes	4	No	30	10x10x10	$5-10000
D-7800	16	10	Yes	Yes	32	Yes	30	Same	$25000
D-7900	16	10*		Yes	32	Yes	30	Same	$27000
D-8000	Mini-computer capable of multiplexing Dynaprobe-7900 series						30		$50000
SUM	16	unk.	Yes	Yes	20	Yes	50	Portable	$21500-50000
CPM-II	16	10	Yes	Yes	20	Yes	50	48x39x49	$43000
X-Ray	32	32	Yes	unk.	96	Yes	unk.	Not Portable	Mini-Computer

* Options available to increase this minimum

Note: cost includes tape unit and monitor unit

Figure 2. Summary of Hardware Monitors

IV. IMPLEMENTATION

The manufacturers' manuals provide the initial guide to implementing a hardware monitor. Generally, a manufacturer will provide manuals on two levels. First, there will be the technical wiring manuals. These, like logic manuals of the mainframe manfacturers, are of little use to the system analyst who is doing the measuring experiments. The second level of manuals is often referred to as the "cookbook" approach manuals. In this type of manual, specific measurements are described to familiarize the user with the actual attachment of the monitor to the host computer system. Some manufacturers suggest in their advertising that their cookbooks will enable the user to measure any problem area and improve the performance of his computer system. No computer center manager should ever be misled by this type of wild claim. The hardware monitor is a versatile measurement device, but it is not a "cure-all" for inefficient computer system operations.

All large computer hardware manufacturers offer numerous operator training courses. The manufacturers of hardware monitors are no exception to this industry trend. It is important to remember that these courses are operator training courses and not courses in how to measure a complex modern computer system. Generally, the courses are of one weeks duration and will ensure that the trainee can push the right buttons and handle minor problems involved in equipment hook up.

Since Mr. D. R. Deese [Ref. 2] suggests that one person should be in charge of the measurement effort, this person is the best choice to attend the manufacturer's training course. He emphasizes that hardware expertise is not required for monitor measurement experiments and that it is much more important for the user to have a solid background in systems design and operations. He will then get a better idea of the limitations, as well as the capabilities, of the hardware monitor. Careful planning of the problem areas to be explored prior to attendance of the training course will enable the analyst to seek answers to the inevitable questions.

Every source of information on hardware monitors points to the lack of system degradation as the key advantage to using a hardware monitor. This advantage can quickly be lost if the daily operations of the computer center are disturbed by open hardware cabinets and probe cables draped over everything in the equipment area. A little extra time is required to lift floor panels or ceiling tiles, but then nobody will trip on probe wire. This not only ruins the measurement experiment, but probes are directly connected to computer contact pins. Breaking a pin on a plugboard can ruin the whole day's production efforts.

Every manufacturer has a library of probe points for the major computer systems. If the person selected to perform the measurement experiments is a qualified system analyst, he will have little trouble finding the probe points in the host system. Major computer manufacturers lay out their hardware contact pins in a matrix fashion. Each

letter or number in the identification number of a probe point is a dimension in the matrix. The individual components of a computer system have these identification numbers permanently affixed to their frames, doors, plugboards, and individual pin locations. The hardware monitor manufacturer will provide procedures to check out probe hook ups prior to measurement experiments.

Nothing will negate carefully planned experiments faster than probe connections on the wrong points. Faulty information from the manufacturer is the least probable, but hardest error to discover. A machine modification or a physical error in probe placement by the user are the more likely sources of errors. Hardware expertise might not be necessary, but basic electrical cautions and practices must be followed. Lack of adequate grounding of probe points is another common source of errors.

Conclusions drawn from measurement experiments should be based on extensive sampling. Short samples can be influenced by unusual jobs, or a device that is malfunctioning. Initial checkout of experiments designed by the user should include multiple measurement of the same activity, since this is an easy form of verifying the measurements.

V. DESCRIPTION OF THE HARDWARE MONITOR AT NPS

This section describes the hardware monitor purchased by this institution and delivered in the summer of 1971. The unit at this computer center consists of two ME-1011 Event Monitors and a ME-2011 Measurement Printer. The hardware monitor is manufactured by Boole & Babbage, Incorporated and is referred to as a Measurement Engine.

Their Measurement Engine product line consists of a variety of hardware measurement tools which can be configured by the user to analyze specific performance problems. The measurement engine components are of modular design which enables the user to expand his system as his needs or desires for more measurement tools increase. This institution is now going through such an expansion process; consideration is being given to purchasing a tape unit to record greater volumes of data.

The controlling module in the Measurement Engine System is the ME-1011 Event Monitor. The Event Monitor uses passive probes to monitor and record electronic signals generated by the host computer system. These signals are then logically combined in a user-determined manner to visually and graphically display the desired measurements. The Event Monitor can be supported by a variety of peripheral output devices, but a Measurement Printer is the only device at this institution.

Each Event Monitor contains six counters with 10**4 count capability, although a 10**6 count capability is optional. Under logic

plugboard control, counters may be cascaded within a single Event Monitor or between multiple Event Monitors. Counters may operate in one of three modes: Percent Utilization Mode (over the specified time periods), Counts per Interval Mode (counting over the specified time period), or Total Counts Mode (counting over externally controlled time period). Counters can operate in any of these three modes individually or collectively.

The Event Monitor is equipped with a removable logic plugboard which allows the user to perform logical operations with measurement probe signals. Logic capabilities include 12 ANDs, eight ORs, 12 INVERTERS, four FANOUTS, and two SET/RESET latches (flipflops). Probe and counter controls also appear on the logic plugboards, which allow activation of probes or setting of counters when a desired signal is sensed. A second plugboard is shown as Figure 3.

Ten customer-selected recording interval ranges are available on each Event Monitor. Ranges are available as multiples of .8333 seconds. Standard intervals are 5 seconds, 15 seconds, 30 seconds; 1, 5, 15, and 30 minutes; and 1, 4, and 8 hours. Custom options are available, but are not on this center's monitor.

Display is by direct readout of percent utilization with autopositioned decimal point, and two-digit interval count; or four-digit display of count with overflow indicator. Buffer storage holds the readings for transmission to recording peripherals.

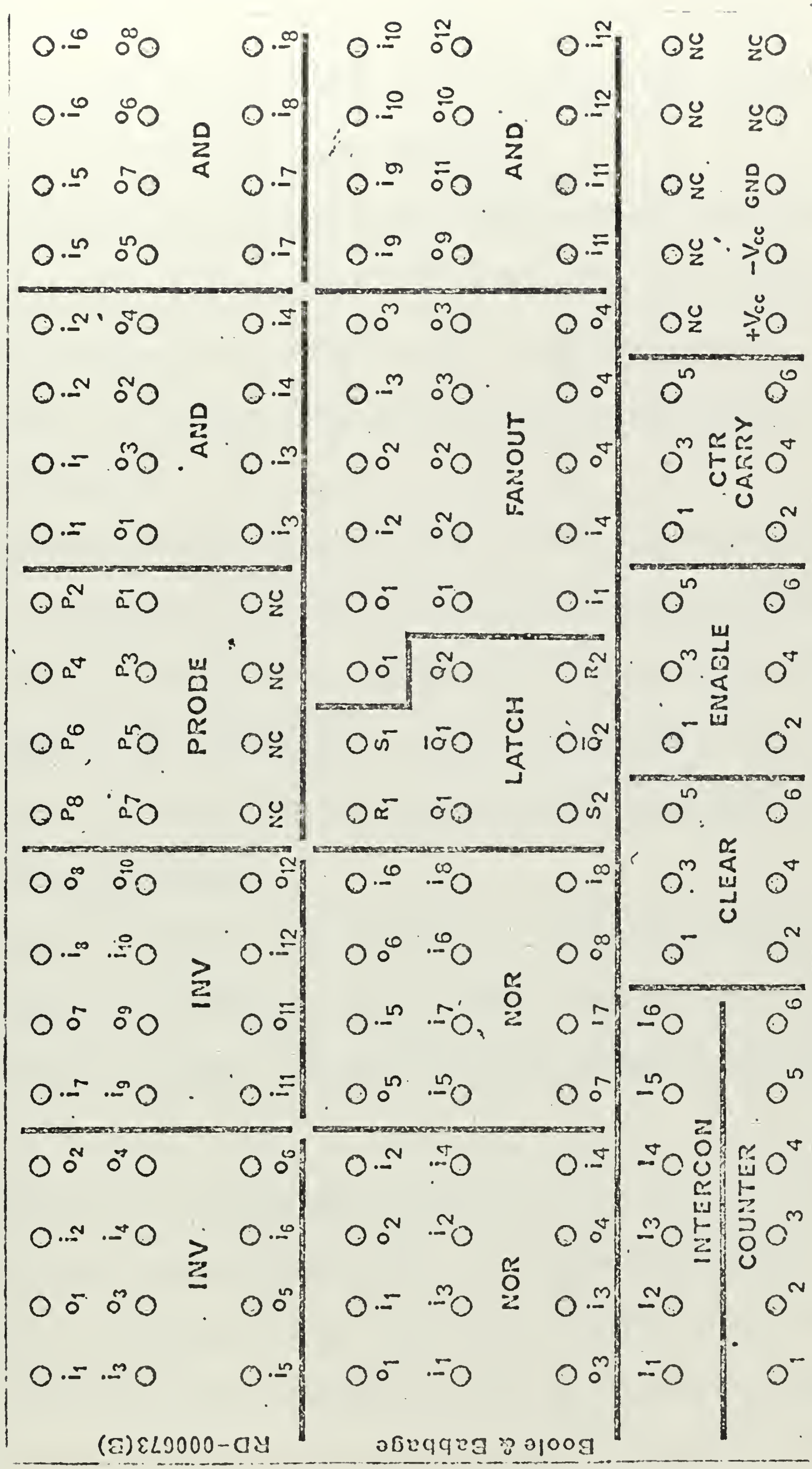


Figure 3. Logic Plugboard

Basic clock frequency is 192.00 KHz. Recording interval is digitally programmed from minimum of 0.8333 seconds up to 2**16 multiples of the minimum recording interval.

A Counter Function switch allows selection of the three modes of counter operation for each counter. Six push buttons allow selection of the specific counter to be displayed on the visual display tubes. The Event Monitor can be reset to the start conditions at any time by a single push button.

Eight ME-2011 Measurement Probes are furnished with each Event Monitor. A maximum of 16 probes can be used by each Event Monitor. Each probe makes three friction connections to wire-wrap pins of the host computer. The connections are signal, signal voltage, and reference voltage. Maximum frequency is ten megahertz and minimum pulse width is 50 nanoseconds.

Each Event Monitor is 16-7/8" wide, 4-1/16" high, and 11-13/16" deep. Each Event Monitor weighs 20 pounds. Multiple Event Monitors stack one upon the other. The face of an Event Monitor is shown as Figure 4.

The ME-2011 Measurement Printer is the sole source of output from the monitor system at this center. It produces a paper tape which is 3-7/16" wide and must be torn from the roll to remove data. Seven lines are printed for each measurement interval, one line per counter, and one line to identify the source Event Monitor. The face of a Measurement Printer is shown as Figure 5.

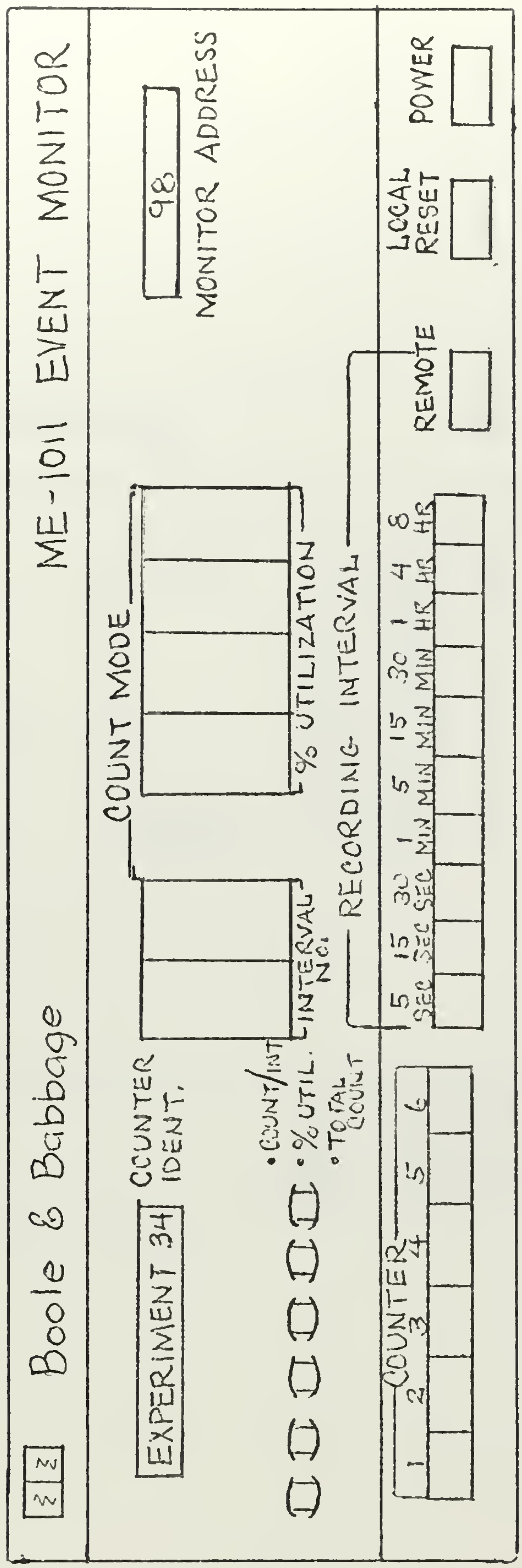


Figure 4. ME-1011 Event Monitor

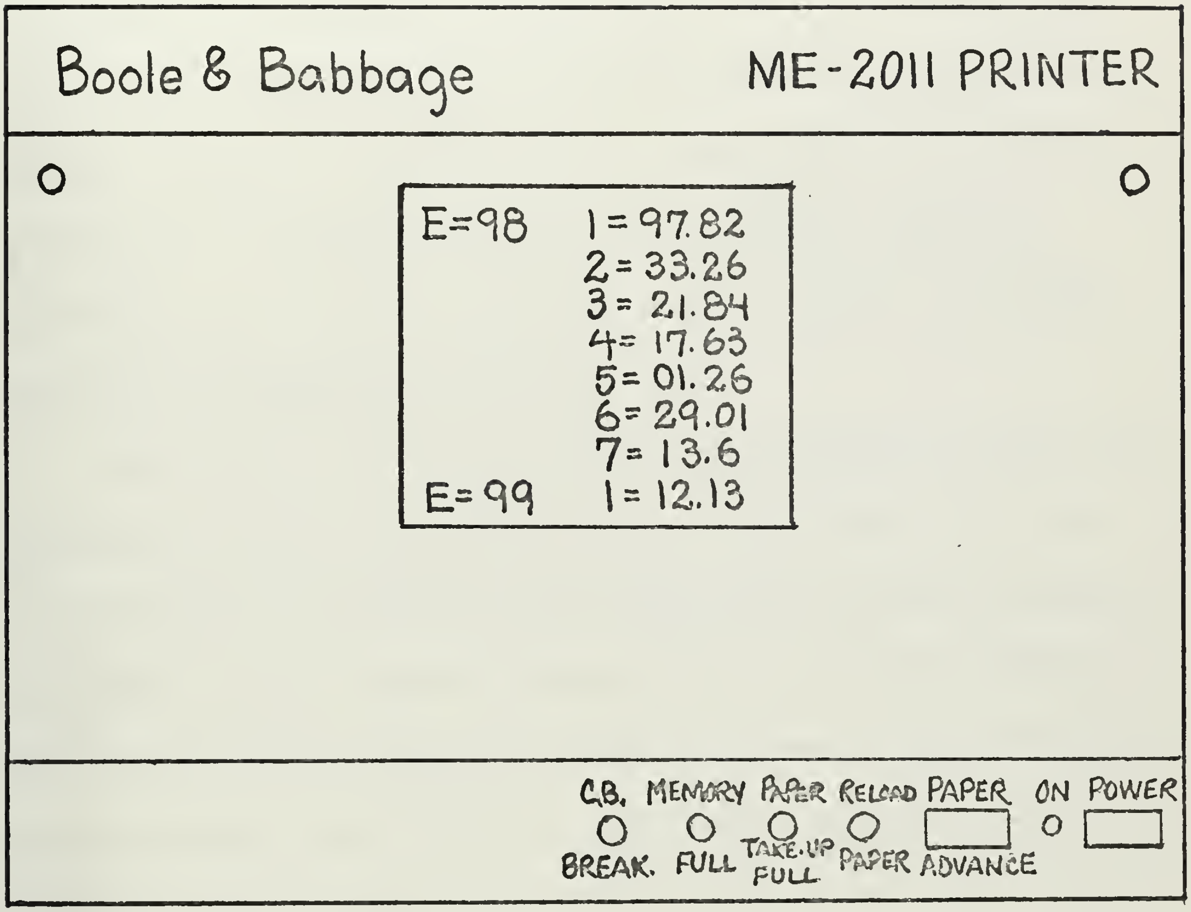


Figure 5. ME-2011 Printer

VI. SYSTEM MANAGEMENT FACILITIES AS A SOFTWARE MONITOR

System Management Facilities (SMF) is an optional feature of the System/360 Operating System that can be selected at system generation in conjunction with Multiprogramming with a Variable Number of Tasks (MVT). SMF collects system and job information as well as providing exits to installation-supplied routines. Although SMF is designed for gathering job accounting information, it also collects significant information to be a fairly sophisticated software monitor, especially when used to supplement a hardware monitor.

SMF gathers statistics on every job step processed for later data reduction by routines supplied by the installation. Jobs can be monitored throughout the system and exits taken when installation-defined conditions are met. Statistics gathered on job and job-step performance can be used by installation-written management information programs reporting system efficiency, performance, and usage. SMF provides control program exits that can be used by installation-written routines to monitor jobs at specific points as they are processed. These routines can enforce installation standards such as: identification, priority, resource allocation, and maximum execution time. Since the need for such statistics and control standards varies so widely, SMF provides a great deal of flexibility. SMF must be specified at system generation time, but its use can be modified at each initial program loading.

The manual, Planning for System Management Facilities [Ref. 5], provides an introduction to SMF concepts, system requirements, and operations. The chapter, System Management Facilities [Ref. 6], of the Operating System Programmer's Guide for the System/360 provides the information necessary to access the actual SMF data set which is located on disk prior to any reformatting according to installation-written routines.

SMF degrades system throughput depending on the options selected by the installation and the efficiency of the exit routines which are also installation-written. This is typical of all software monitors.

VII. SYSTEM PERFORMANCE PROFILE

Bonner [Ref. 4], Boole & Babbage [Ref. 7] , and Cockrum and Crockett [Ref. 8] all suggest that the first performance measurement experiment should be a system profile. A system performance profile is essentially the measurement of the average activity level of system components. This section details the indicators of a system performance profile and what corrective action should be taken to achieve a more balanced system.

The logic capabilities of the hardware monitor are used to measure CPU and I/O overlap, CPU and channel active, selector channel active only, CPU active only, CPU wait state and channel active, and activity of the individual major components. The objective of this type of experiment is to seek components that are either overworked or under utilized. The rationale for multiprogrammed computer systems is to make better use of the system resources by having more than one job step active at a time. The system profile experiment is designed to measure this utilization of system resources.

Now that several experiments have been completed, the hardest job is ahead of the user. The results of the experiments must be analyzed and recommendations must be made to achieve the desired objectives. Cockrum and Crockett [Ref. 8] discuss the basic indicators of system utilization that the user should look for in his experimental measurements. Much of the following paragraphs is taken from their paper.

The basic indicators to look for in interpreting the system performance profile are small channel overlap, channel imbalance, high channel utilization, large wait only, and large CPU active only. Each of these indicators will now be discussed in greater detail along with possible solutions to the indicated problem area.

The probable reason for small channel overlap, even when the channel utilization is high, is poor device placement on the channels. This results in sequential operation of the devices as a job step executes and requires these devices. This indicator suggests that the control units and devices should be monitored to determine which devices and data sets should be moved. A new system profile would then be taken to verify the expected results of a system configuration change. The configuration simulator which is offered with the SUM monitor system would be an excellent way to check out such proposed changes without adversely affecting the daily production requirements placed on a computer system. As a side note, both Cockrum and Crockett are employed by Computer Synectics, who manufacture SUM.

A small channel overlap when the utilization of the channels is low is a prime indicator that all the work could be placed on one channel without adversely affecting the processing of system work. The user must be careful that the period or periods he used to conclude that activity was low, were not unusual. This is about the time that the user of a hardware monitor begins to see the need for real time information on the system load during monitoring.

If the channel utilization is high, but the channel load is not balanced, the device activity needs to be measured to determine which devices should be moved. Again, any reconfiguration must be verified by another system profile. The case for continued monitoring after a re configuration cannot be overemphasized. Changes in the basic job stream and any modifications to operating systems or even operating procedures can have startling effects on the performance of a modern computer system. Nobody can ever claim to understand the implications of any change made in the environment in which the computer system operates.

Low channel utilization and channel imbalance would indicate that all the work could be placed on a single channel. Yes, hardware monitoring may reveal that you do not really need new equipment and your old system is not being utilized to anything approaching its capacity. A side benefit of monitoring is checking that system components are performing as the manufacturer advertised they would. A card reader that is not quite up to rated input rates can slow down the slowest part of any computer system.

High channel utilization indicates that system data sets should be examined. There may be a problem as to which routines are resident in core and which are maintained as non-resident. A measurement should be made to determine transfer time into core of system routines relative to device activity. If the transfer time is high and the current devices are not going to be replaced with higher speed devices, then make all

system routines non-resident and measure their activity to determine which routines should be resident. Such experimenting with system configuration is essential if improvements are to be obtained. Unfortunately, there is no universally best method to operate a computer system and achieve efficient performance at a minimum cost. Another good thing to remember is that maximizing the performance of a system and the reduction of operating costs are often competing interests. Efficient operation of the installed equipment is most likely the goal of most system performance monitoring today.

Another possible cause for high channel utilization is record blocking in data sets on direct access devices. Measurements of I/O device utilizations and examination of the data sets on each device should be made to determine data sets in which a larger number of records could be placed in each block to increase the efficiency of access.

If the system performance profile shows a large amount of wait only time for the CPU, the (disk arm) SEEK-only time should be measured. If a large portion of the wait and no channel busy time is SEEK-only time, this indicates the system is waiting for seeks on the direct access devices. The direct access devices should be measured to determine which data sets are poorly placed and thus causing the arm contention on the direct access device. The console log can be used to correlate seek times with programs active. This correlation can be a help in determining that a particular partitioned data set has excessive arm

movement between the sequential sets. Seek time can be reduced by rearranging data sets on the same disk pack or moving data sets to different disk packs.

If the SEEK-only time is insignificant, operation problems are indicated. Possible causes are difficult operator set-up procedures, too few operators, poor job scheduling, etc. Measurements should be made as to the amount of not ready time for each device during the day. If a large amount of not ready time is discovered, operation problems or equipment malfunctions are indicated.

Large CPU active only time and a low CPU-Channel overlap indicate that effective multiprogramming is not taking place. This does not indicate that the computer is not capable of multiprogramming. The job stream may not contain the balance of computer and I/O bound jobs needed to take advantage of multiprogramming. Improper location of data sets can force the most powerful multiprogramming system to spend all its time searching for the required data sets. Most university computer systems are presented with programs that are just plain inefficiently written. The beginning computer programmer cannot and does not consider making his programs conducive to multiprogramming. Care must be taken that system performance profiles are gathered using job streams that reflect the typical job stream of the host computer system.

The following section describes the system performance profiles obtained at this institution. The indicators of system balance are presented and corrective action is recommended to balance the system.

Dr. G. Carlson [Ref. 9] lists some typical, very preliminary measurements for an IBM 360 installation. These measurements are presented here as Figure 6. Later in this work the observed values for similar preliminary measurements for this institution's IBM 360 are tabulated.

Event	Percentage Active
CPU active (no slow speed bulk core)	20-50%
CPU active (with 4X slower bulk core)	40-70%
Selector channel active (disks)	20-40%
Selector channel active (tapes)	2-15%
Multiplexor channel active	0- 5%
Console typewriter	10-20%
Large core storage busy	10-25%
Supervisor state	25-40%
Supervisor state as a percentage of CPU busy	40-60%

Figure 6. Preliminary Measurements for an IBM 360

VIII. EXPERIMENTS PERFORMED

When the Boole & Babbage Measurement Engines and Printer were delivered to this institution, there were very few of them in existence. This firm is noted for its software monitors and has just recently expanded into the hardware monitor manufacturing business. They now offer a complete measurement package to their customers. The manual received with this unit was of the first level discussed in this paper -- it was an engineer's manual of how the box worked. A cookbook type of manual was on the way, but had not yet been completed.

The receipt of the second level manual [Ref. 3] did not open new and easy avenues to the measurement of the 360 at this institution. Rather, it provided a detailed account of how to implement the system profile measurement described in section VII. Probe points for the IBM 360/65 (which are a subset of those of the 360/67) are identified and their location in a typical system layout is pinpointed. The configuration for the logic panel of the measurement engine for each suggested experiment is presented in a standard electrical engineering diagram of the logic gates and their connections. This is not of great value when trying to configure the logic board of the monitor. A form is used at this institution, which is a diagram of the logic panel with no connections made, to plan proposed experiments. This form is shown as Figure 3 in section V, and is much easier to use than the logic diagrams.

The current edition of the applications manual for the Measurement Engine contains several other experiments for a System 360. Those experiments which were performed are explained in detail later in this section.

Two of the earlier system performance profiles were invalidated by accidents, which are typical of the problems that an institution will face with its first hardware monitor. The paper jammed in the printer shortly after a new roll of paper was inserted. This was good enough to necessitate a call for help to the manufacturer and several days lost time. One later experiment designed to obtain a typical days activities was thwarted when the operator turned the Measurement Engine on, but did not turn the printer on. This was just a lack of good communication between the monitor user and the computer system operator as to what the former wanted done. Making measurements which verify themselves saved this institution from performing experiments with a faulty monitor. It is always necessary to eliminate the possibility that the monitor malfunctions and produces incorrect results before any actual measurements are made. The fault was quickly repaired and the Measurement Engine was back in operation.

A. EXPERIMENT 1

The first successful system performance profile was conducted over a four hour period using a selected interval of fifteen minutes. The hardware monitor integrates the percent active over each interval for each

counter. Every fifteen minutes the printer records the percent utilization and the monitor resets, ready to measure the next interval.

The IBM Model 67 configuration is shown in Figure 7. Figure 8 indicates how resources are split when CP/CMS time sharing is active, which is 1200-1600 hours on week days. The drum storage is used by OS, when CP/CMS is not operating, for the Quickrun system. This is designed to obtain higher paging rates and reduce system overhead.

The three probes used were: CPU manual mode, CPU in wait state, and Selector Channel two active. The actual pin locations probed are listed in Appendix C on the diagram of the logic patchboard used in this experiment.

Six events were measured by the logical combination of the signals from the three probes listed in the preceding paragraph. These events were: Computer not in manual mode, CPU in wait state, CPU in wait and Selector Channel two busy, CPU active and Selector Channel two busy. Three additional events were calculated by the program Hardware Graph. These events were: CPU active, Selector Channel two only active, and the CPU only in wait. (Selector Channel two only active means the Selector Channel two active AND CPU in wait state. This indicates system is waiting for I/O completion. CPU only in wait means both the CPU AND Selector Channel two are inactive. This indicates no system activity.)

The three events CPU active only, Selector Channel two active, and CPU in wait only were used as a check on the measurements taken.

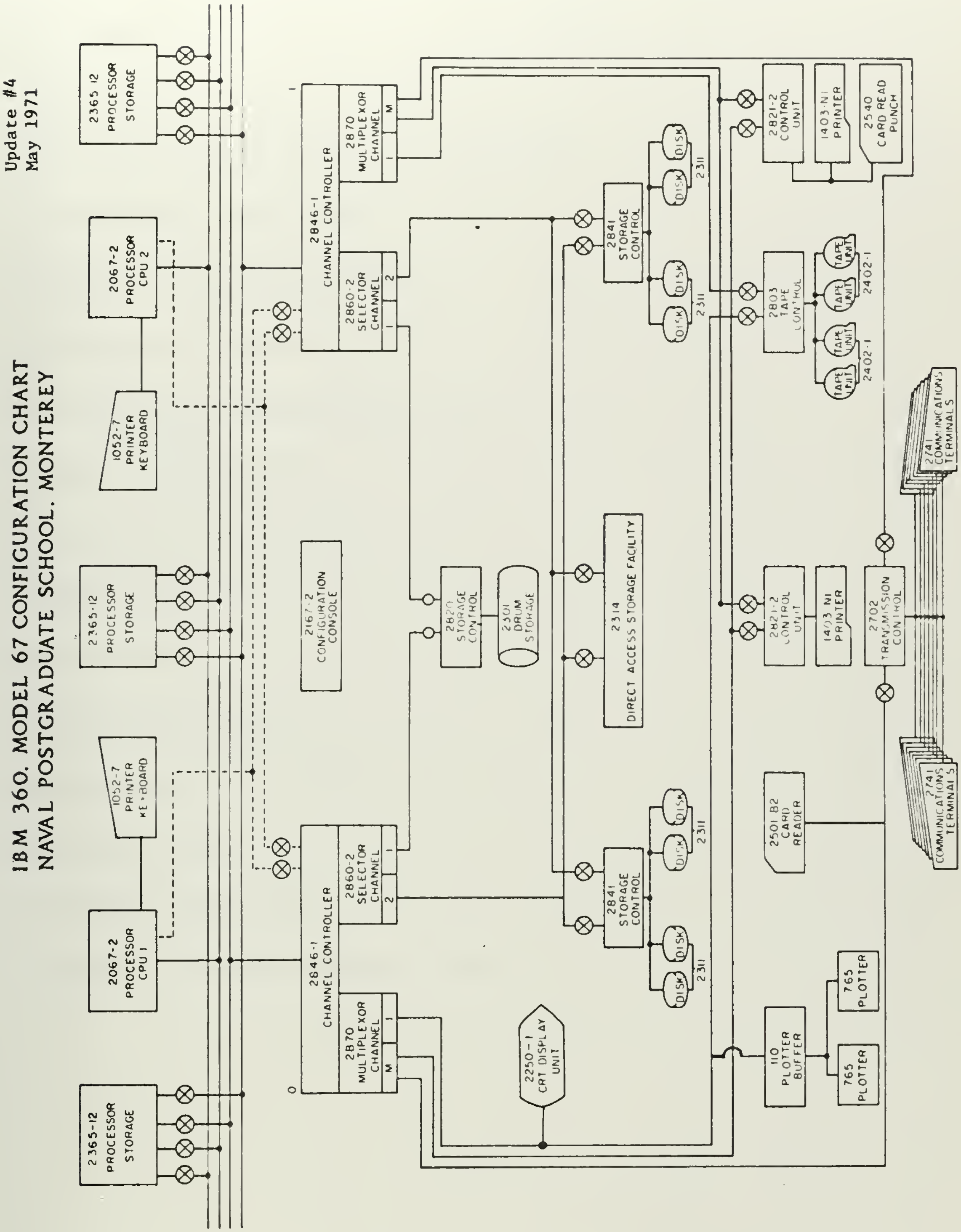


Figure 7. Naval Postgraduate School IBM 360 Model 67

DEVICE	OS	CP/CMS
CPU 2067-2	X	
CPU 2067-2		X
PRINTER KEYBOARD 1052-7	X	
PRINTER KEYBOARD 1052-7		X
PROCESSOR STORAGE 2365-12	X	
PROCESSOR STORAGE 2365-12	X	
PROCESSOR STORAGE 2365-12		X
DRUM STORAGE 2301		X
DISK STORAGE 2311 (8)		X
DISK STORAGE 2314	X	
TAPE UNITS 2402-1	X	X
CARD READER 2501-B2	X	
CARD READ PUNCH 2540		X
PLOTTER 765 (2)	X	
PRINTER 1403-N1 (2)	X	X
CHANNEL CONTROLLER 2846-1 (2)	X	X

Figure 8. Computer Resource Allocation Under CP/CMS

(CPU active _only_ means CPU active and Selector Channel two inactive. This indicates CPU is computing while I/O is inactive.) They should total 100%. This reveals that the intended probe points are probably the ones that are being measured.

Figure 9 is an example of the graphic output that Hardware Graph, which is listed in Appendix A, produces. Hardware Graph produces graphs for each of the nine events being monitored in a system profile. Most data reduction packages produced by the manufacturers of hardware monitors produce similar graphs of the event utilization trends during the experiment. Such packages also produce tables showing the actual data gathered by the monitor. Figure 10 shows both a graphic representation of the percent utilization for the event being measured and the actual raw data on the end of each bar of the graph. It will be noted that 100% is represented by 99.99%. Such graphic presentation of the raw data is far superior to the long tape of four digit numbers produced by the monitor's printer. See Figure 5 for an example of the printer tape output. An improvement could be made in this data presentation by presenting tables for each interval showing several events at once. This could easily be implemented to fit the needs of the individual experiment.

Figure 10 is an example of a system performance profile taken from the first experiment conducted. Each fifteen minute subinterval produced a system performance profile. The program Hardware Graph showed trends in each of the nine performance indicators on a separate

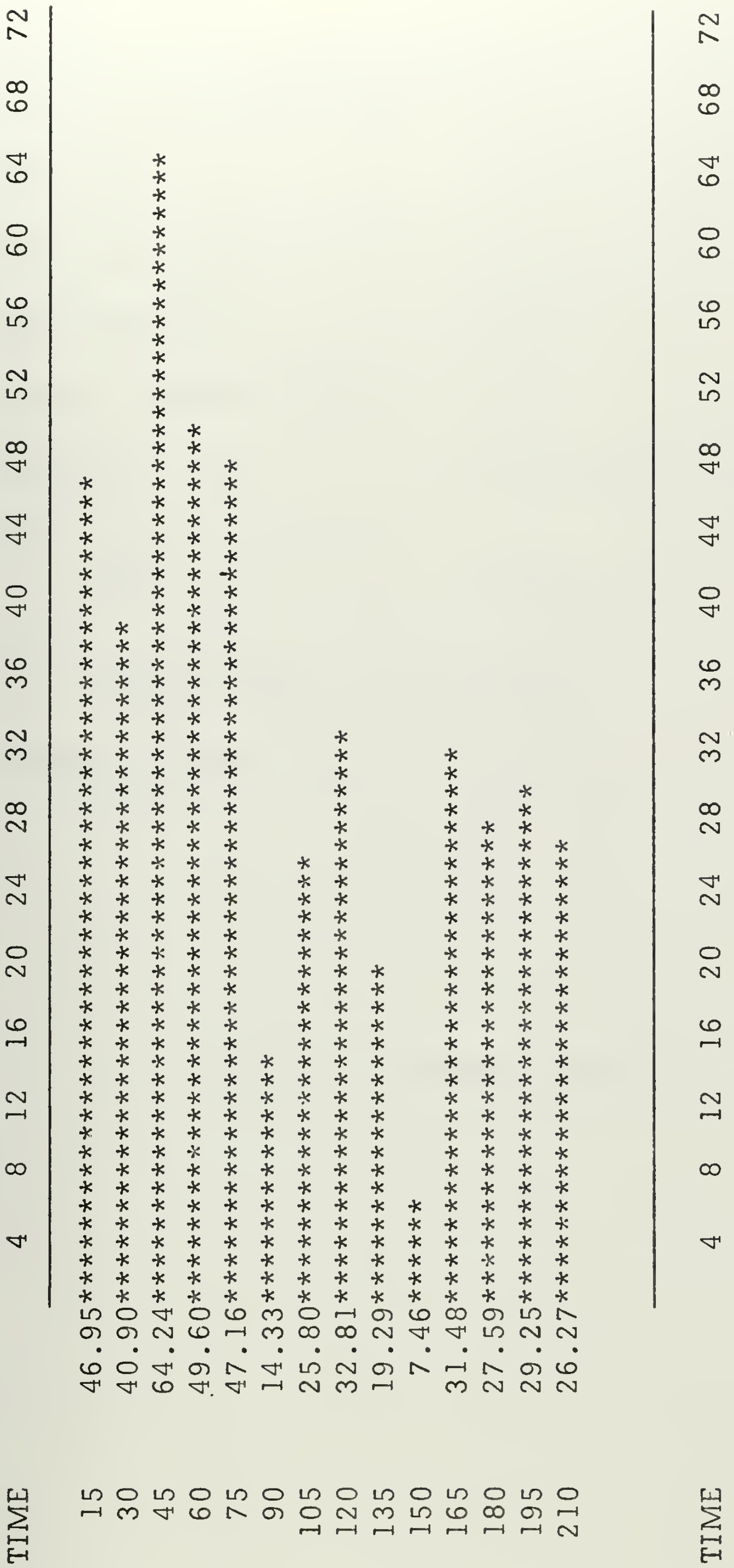


Figure 9. Graphic Output from Hardware Graph

EVENT	PERCENTAGE ACTIVE
Machine not Manual Mode	99.99
CPU Active	47.06
CPU Wait state	53.93
CPU Wait only	23.47
Selector Channel 2 Busy	48.93
CPU Active only	27.59
CPU Wait and Channel 2 Busy	29.45
CPU Active and Channel 2 Busy	19.48
Selector Channel 2 Busy only	19.48

Figure 10. System Performance Profile

graph, like Figure 9. The right hand part of the graph shows which items should be added together to make a system profile of 100%, thus indicating the system tradeoffs. It should be noted that the events CPU Active and channel 2 Busy, which is obtained logically, is the same as the event Selector Channel two Busy only which is calculated from other events.

Figure 11 is the range of results from this first experiment. The raw data was smoothed, highest and lowest percentages were eliminated, to produce this figure. The measurements were made from 1000 to 1400 hours, which were the busiest hours of operation. (This fact was obtained from the monthly utilization reports published by the computer center staff.) Such peaks of system activity can often be obtained from the System Management Facilities (SMF) reports, thus saving needless monitoring during periods of low activity.

B. EXPERIMENT 2

Figure 12 shows the results from system performance profile number two. After experiment one, it was thought that a fifteen minute measurement interval might be too long to measure the fluctuations of system performance. A fifteen second interval was used and measurements were taken beginning at 1300 hours. Twenty-four fifteen-second intervals were recorded. The results in Figure 12 indicated that the system was not under heavy load at the time. The anticipated fluctuations did not appear.

Event Name	Percentage Range
Machine not Manual	99-100
CPU Wait state	30- 70
CPU Wait state and Channel 2 Busy	5- 30
CPU Active and Channel 2 Busy	10- 30
CPU only Active	25- 30
Selector Channel 2 Busy	30- 50
CPU Active	30- 70
Selector Channel 2 only Busy	12- 30
CPU Wait state only	15- 40

Figure 11. Summary of System Profile #1

Event Name	Percentage Range
Machine not Manual	99-100
CPU Wait state	40- 80
CPU Wait state and Channel 2 Busy	25- 40
CPU Active and Channel 2 Busy	9- 24
CPU only Active	12- 29
Selector Channel 2 Busy	40- 55
CPU Active	25- 48
Selector Channel 2 only Busy	12- 20
CPU Wait state only	25- 40

Figure 12. Summary of System Profile #2

C. EXPERIMENT 3

Figure 13 shows the results of the third system performance profile experiment. This experiment started at 1400 hours and used a thirty second recording interval. Twenty-five intervals were recorded, which is the maximum number of intervals that the subroutine in Appendix A can currently produce. (The size limitation is due to array dimensioning and the practicality of getting the graph on one page of computer output.) It appeared in experiment three that the computer was spending a lot of time waiting on the completion of input/output operations by the selector channel.

D. DEVELOPMENT OF SMF GRAPH

It was at this point that it became apparent that information on the job stream on the machine during experimentation must be obtained. The program, SMF Graph, which is listed as Appendix B, extracted the needed data on job stream activity. No software monitor was added to the system, thus the system was not degraded further by a software monitor during measurement experiments.

The current SMF file is read and key information is accumulated about the job steps executing (actually being terminated) during the measurement period. Care must be taken that the SMF does not complete a file and switch to the alternate disk during the experiment or all SMF data will be lost. Once a file has been filled it is dumped to a system program that compacts it and writes it on tape. It is still available to the user, but the program SMF graph will not be able to read it.

Event Name	Percentage Range
Machine not Manual	99-100
CPU Wait state	52-85
CPU Wait and Channel 2 Busy	30-55
CPU Active and Channel 2 Busy	8-20
CPU only Active	7-17
Selector Channel 2 Busy	45-70
CPU Active	15-30
Selector Channel 2 only Busy	8-12
CPU Wait state only	20-40

Figure 13. Summary of System Profile #3

Generally, this switch occurs during the midnight shift of computer operations.

SMF records fourteen different types of records. Reference 6 details their format and contents. Record type four is the job step termination record. This is the one used by this research and is the source of job stream data for SMF Graph.

Ideally, one would want to gather statistics on only the resources expended during the period being monitored, but if the entire experimentation period, in this case eight hours, is broken into large subintervals, then job steps terminating during this subinterval are a good indicator of system activity. Thirty minute subintervals were chosen because the earlier experiments showed little fluctuation in system activity with smaller intervals.

SMF Graph keeps separate statistics on nine categories of job step terminations. It does this by extracting the name of the system program executed by the job step, Fortran G compiler, GPSS, WATFOR, etc. and then comparing this name with the nine types specified by the user in variable T1-T9 in its declarations and initializations. The program then branches to a section and records the desired statistics on this particular job step termination record. The current version of this program gathers the following data: initiation time, termination time, program executed, record type, and CPU seconds used. The user can easily change the values of T1-T9 and record the above statistics on any system program that he is interested in obtaining data upon. The nine current programs being monitored are: WATFOR, WATFORC, ALGOL, FORTRAN G COMPILATIONS, FORTRAN LINK STEPS, FORTRAN GO STEPS, FORTRAN H COMPILATIONS, GPSS, and QUICKRUN. It was determined from the monthly center usage reports that these categories made up more than 80% of the jobs being submitted.

Output from SMF Graph includes graphs in the same format as Figure 9, which show for each subinterval the number of steps that executed each of the nine system programs listed above. Graphs are also presented that indicate what percentage of the total system time used during the subinterval went to each of the nine program types. In a multiprogramming system, if one records the initiation to termination time of all the job steps executed during an interval and divides it by the actual clock time elapsed during the interval, an approximation to

the number of job steps active at one time can be determined. As an example, if job A uses 5 seconds, job B uses 5 seconds, and job C uses 10 seconds of system time during a 10 second interval, there were two jobs active at all times, since twenty seconds system time was used in a 10 second interval. Tables at the end of the graphic output of the program indicate the number of job steps active at one time in each interval. Figure 14 is an example of this output.

Average number job steps was 3.560 during interval 1

Average number job steps was 2.746 during interval 2

Average number job steps was 3.807 during interval 3

Average number job steps was 0.198 during interval 4

Figure 14. Job Steps Active at One Time

Tables are also printed for each subinterval which indicate the number of programs executed in each of the nine program types. If the user wished to change the nine program types now being recorded, the titles for the graphs would have to be changed (they are read in as data) and the formats for the tables would have to be changed to the names of the new system programs being monitored. Figure 15 is an example of the tables output by SMF Graph. The variable Core-Use which appears in these tables is the ratio of system time used to CPU seconds used for each of the nine system programs. Programs with a large Core-Use spend a lot of time in core in order to obtain very little CPU time.

Job steps recorded in interval 7

WATFOR job steps equals 0	core use 0.0
WATFORC job steps equals 0	core use 0.0
ALGOL steps equals 0	core use 0.0
FORTRAN G compile steps equals 12	core use 12.867
FORTRAN LINK steps equals 19	core use 69.049
FORTRAN GO steps equals 18	core use 46.168
FORTRAN H compile steps equals 0	core use 0.0
GPSS steps equals 0	core use 0.0
QUICKRUN steps equals 17	core use 26.657

Figure 15. Job Steps Recorded in Each Interval

The SMF Graph program requires certain input data to ensure that its output will parallel the output of the hardware monitor printer. The length of each subinterval in seconds, the number of intervals being monitored, and the starting time of the measurement experiment must be input. The titles that appear on the top of the 18 graphs must also be read in.

The period from 0800 until 1600 hours was selected as the interval to be monitored. This interval was broken into thirty minute subintervals. Due to difficulties in getting the hardware monitor attached to the computer system, six days of SMF data was gathered for the 0800-1600 period before the monitor was ready. This did establish the normal workload of the system and vindicated certain assumptions that had been

made. A summary of this workload is presented in Figure 16, which shows the average 30 minute interval for each of the six days.

The most striking fact to be seen in Figure 16 is the small number of job steps that terminate during a one half hour period. Operating policy at this institution prevents large jobs from tying up the computer resources during the 0800-1600 time interval that these statistics were computed from. Day 1 on Figure 16 shows the fewest number of job steps terminated on the average one half hour period. This is due in part to hardware malfunctions which necessitated the curtailment of normal operations. Days 2 and 3 are the weekend and fewer jobs are run from 0800-1600 on the weekend than during the normal week days. It appears that about forty job steps terminating each half hour is the average workload for this computer center. It should be pointed out that the WATFOR type jobs are now running under QUICKRUN, which accounts for the small number of WATFOR type steps in the averages of Figure 16.

E. EXPERIMENT 4

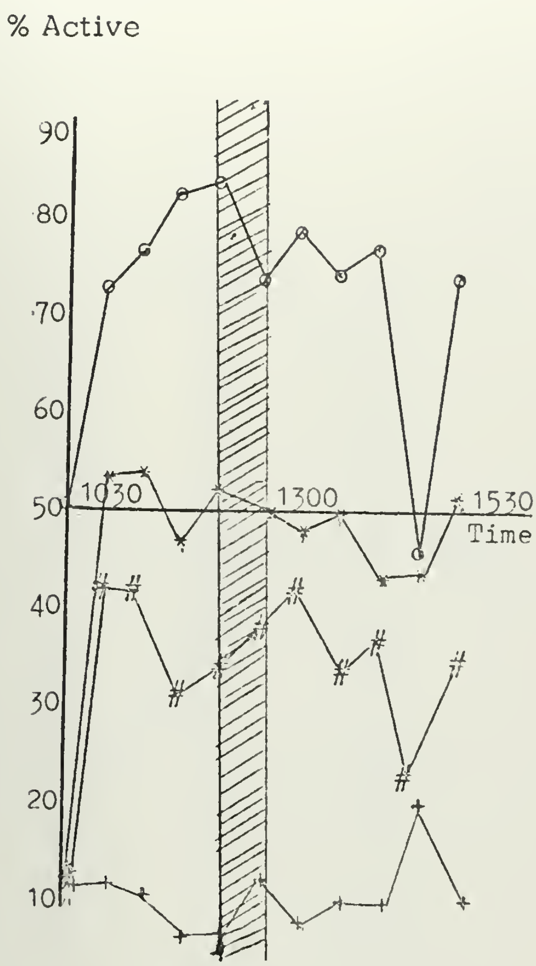
Two days during the week were used to gather hardware monitor data and SMF data concurrently. The time interval was from 0800-1600 and the subinterval was thirty minutes. Figures 17 and 18 show the trends of four basic system performance indicators during the experiments. The four basic indicators are: CPU Wait state, Selector Channel 2 busy, CPU Wait state and Selector Channel 2 busy, and finally, CPU active and Selector Channel 2 busy. The last two indicators are representative of the time the CPU waits for a busy channel and the amount of CPU and

Activity	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Ave.
Number of							
WATFOR STEPS	0.44	0.18	0.37	1.16	0.37	0.56	0.51
WATFORC STEPS	0.00	0.06	0.25	0.33	0.12	0.18	0.16
ALGOL STEPS	0.56	1.06	0.18	0.16	0.25	0.44	0.44
FORTRAN G COMPILES	8.06	9.06	4.44	10.60	7.62	7.56	7.89
FORTRAN LINKS	8.09	12.10	7.50	13.02	10.22	10.34	10.19
FORTRAN GO STEPS	7.88	11.41	7.87	12.83	10.04	10.11	9.98
FORTRAN H COMPILES	0.00	0.06	0.50	0.00	0.00	0.00	0.10
GPSS STEPS	0.50	0.53	1.56	1.16	2.00	1.18	1.15
QUICKRUN STEPS	unk.	unk.	4.31	25.10	15.21	16.22	15.21
Percentage System Time * Used							
WATFOR STEPS	2.06	0.03	0.87	0.55	1.87	0.45	0.94
WATFORC STEPS	0.00	0.01	0.01	0.00	0.00	0.00	0.00
ALGOL STEPS	1.31	2.94	0.37	0.21	0.31	0.78	0.98
FORTRAN G COMPILES	17.51	13.33	5.52	8.34	13.32	8.01	10.91
FORTRAN LINKS	18.80	15.01	8.06	16.51	11.20	12.54	13.71
FORTRAN GO STEPS	28.90	46.01	20.03	17.31	23.30	22.32	26.33
FORTRAN H COMPILES	1.60	0.80	0.87	0.00	0.00	0.00	0.54
GPSS STEPS	5.87	0.68	5.44	1.45	2.12	1.87	2.90
QUICKRUN STEPS	unk.	unk.	6.68	34.10	29.31	27.81	24.44
STEPS ACTIVE AT ONCE	2.24	2.50	1.69	2.82	3.20	2.91	2.56

Note: All figures are averages for a half hour period.
* This indicates core resident time.

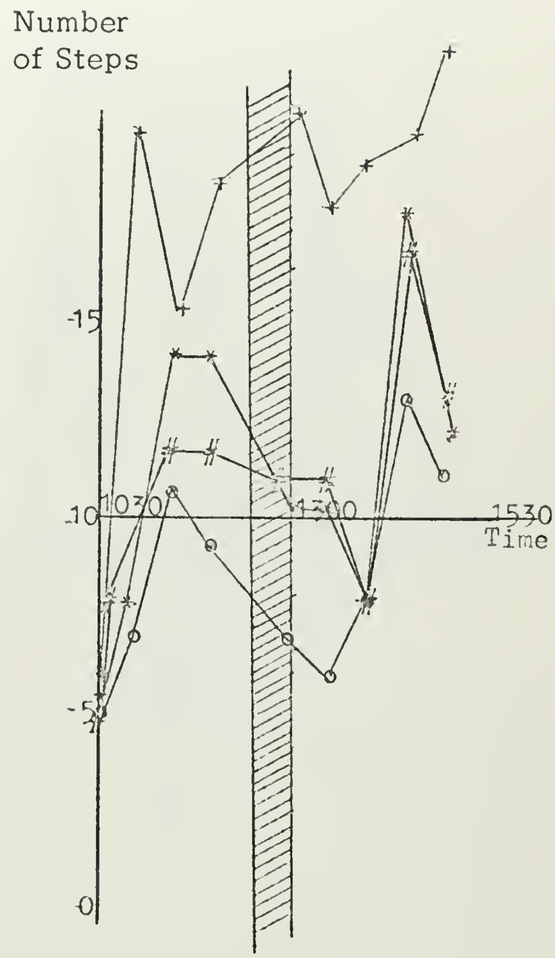
Figure 16. Summary of Workload

Hardware Activity



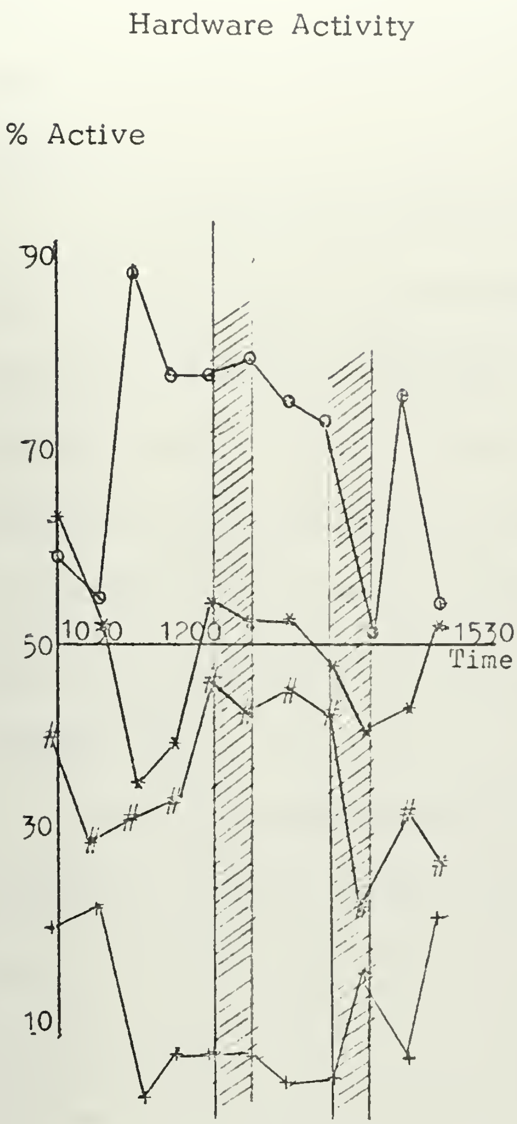
o CPU Wait
* Channel 2 Busy
# CPU Wait and Channel 2 Busy
+ CPU Active and Channel 2 Busy

Job Stream

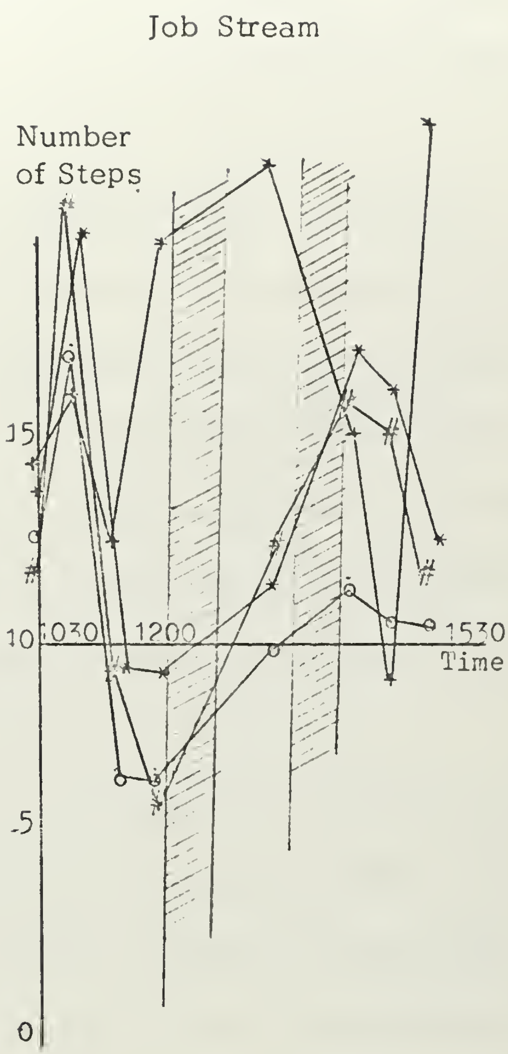


o FORTRAN G Compilation
* FORTRAN Link
# FORTRAN Go Steps
+ QUICKRUN Steps
Hardware Problem

Figure 17. Trends of Performance Indicators 1



- o CPU wait
- * Channel 2 Busy
- # CPU wait and Channel 2 Busy
- + CPU Active and Channel 2 Busy



- o FORTRAN G compilations
- * FORTRAN Link Steps
- # FORTRAN Go Steps
- + QUICKRUN Steps
- Hardware Problems

Figure 18. Trends of Performance Indicators 2

channel activity overlap, respectively. Figures 17 and 18 indicate the job stream activity in addition to the performance indicators. This enables the analyst to see if a particular system program degrades system performance.

On both days the system was practically inactive before 1000, thus Figures 17 and 18 do not begin to show system activity until the end of the 1000-1030 subinterval. Unfortunately, the computer system suffered hardware failures on both days. This resulted in less data to record, but it showed that the computer system performance after a hardware failure was the same as performance before the hardware failure. The card reader is shut off during hardware failures and no jobs are backlogged.

The CPU appears to spend between 70 and 80% of the time in the wait state. Selector Channel number two is active about 50% of the time. The CPU is waiting for the channel to complete its work about 40% of the time. This is preliminary indication that the placement of all direct access devices on selector channel two should be reconsidered. Further action is recommended in section IX to explore this problem area.

A multiprogramming system is designed to overlap the execution of CPU resources with input/output operations by the selector channels. It can be seen from Figures 17 and 18 that this is occurring only about 10% of the time the system is operating. Section IX discusses some reasons for this low utilization of the multiprogramming feature of this computer.

It is clear that even with six days of SMF data on the job stream and two days of hardware monitoring it is difficult to tell if the computer system is operating efficiently. Suggestions are made in section IX of this work to continue the joint collection of SMF and hardware monitor data.

Data from the program Hardware Graph and the program SMF Graph were used to get some idea of the amount of CPU time that was being used for tasks other than the execution of job steps. This CPU time will be referred to as Overhead. The percentage of CPU active time in a four hour period in each day of Experiment 4 was obtained from Hardware Graph, from which the number of CPU seconds actually used to process job step executions was calculated. SMF Graph provided the data to calculate the number of CPU seconds used to process each of the nine program types previously specified by the author as well as the amount of CPU seconds used to process other program types, which meant all job steps execution time was calculated. The difference between total CPU active seconds and CPU seconds used on job step executions was Overhead. Percentage of CPU seconds devoted to Overhead was calculated by dividing Overhead by the CPU active time during the four hour period. Figure 19 shows the Overhead percentage and percentage of CPU active seconds used for each of the nine program types during the two days of Experiment 4. Data from the periods of hardware failure has been eliminated from these calculations.

DAY ONE

Overhead = 42.7%

Program Type	Percentage CPU Active
WATFOR	0.15%
WATFORC	0.14%
ALGOL	0.49%
FORTRAN G COMPILATIONS	8.21%
FORTRAN H COMPILATIONS	0.0%
FORTRAN LINKS	2.17%
FORTRAN GO	20.97%
QUICKRUN	12.08%
GPSS	2.18%
OTHER TYPES	10.36%

DAY TWO

Overhead = 39.1%

Program Type	Percentage CPU Active
WATFOR	0.05%
WATFORC	0.02%
ALGOL	0.74%
FORTRAN G COMPILATIONS	13.62%
FORTRAN H COMPILATIONS	0.0%
FORTRAN LINKS	2.13%
FORTRAN GO	13.17%
QUICKRUN	7.94%
GPSS	5.03%
OTHER TYPES	19.00%

Figure 19. CPU Utilization (Experiment Four)

Appendix D contains the graphic output from day one of Experiment 4. Samples of the data collected by Hardware Graph and SMF Graph are presented.

IX. CONCLUSIONS AND RECOMMENDATIONS

There is a real need to measure the performance of a modern computer system prior to changing its configuration, in order that the changes will most improve the performance.

Any computer center manager who purchases a hardware monitor to simply monitor his computer will be sadly disappointed by the results. No answers to his operational problems will be solved by the hardware monitor. Prior to purchasing a monitor he must prepare a program for measuring system performance and have it firmly established. A highly qualified systems analyst, who knows the present operations in detail, is the ideal man to be in charge of the monitoring program. He may need help to connect the hardware monitor, but the analyst should conduct or direct the measurements of system performance.

Most analysis packages, data reduction programs, or report generators are simple graph and table producers which do very little reducing and a lot of presenting of the counter outputs from the hardware monitor. The staff at any modern computer center could easily produce reports that are specifically tailored to the needs of that computer center. The data analysis packages that come from the hardware monitor manufacturers are generally not very expensive. This is more than likely because these programs do not do very much.

It appears that the management of a computer center can get immediate improvements in system performance by keeping the users informed of

the need to use the computer efficiently. At this institution it was noticed during the research that when a FORTRAN job fails to compile successfully, it does not terminate its use of resources on the link and go steps. System resources are still used to execute these steps; they are in the system, using core and channels, when there is no need for them to be there. A short range cure for this problem is better communication. Management should encourage users who are just developing a program to compile rather than compile, link, and go. A longer range cure would be the modification of the operating system to skip link and go steps that are to follow compilations that have failed. This may not even be possible, but it does seem like a waste of valuable system resources to load job steps that have no chance of executing.

Earlier it was stated that the nine program types, on which SMF Graph collected statistics, represented about 80% of the jobs submitted at this computer center. This fact was obtained from the monthly reports of the computer usage that are prepared by the computer center staff. Investigation of these reports for the last six months shows which languages used the most computer resources. More than 60% of the jobs were either WATFOR or FORTRAN. These jobs used about 60% of the CPU resources for the month (September 1971). The WATFOR type jobs comprised 26.8% of the jobs, but only used 1% of the CPU seconds. This is due to the size of WATFOR jobs, and the efficiency of the WATFOR system in handling jobs requiring limited resources. Other FORTRAN jobs comprised 32.9% of the jobs and used 62.6% of the CPU resources. Many of these

jobs are small enough to run under WATFOR. A very limited test was made to get an approximate idea of the savings encountered when running small FORTRAN jobs under the WATFOR system when compared to using the FORTRAN G compiler. A typical job that executed in under eight seconds using the FORTRAN G compiler would execute in less than two seconds using WATFOR. (This is typical of results obtained at other 360 installations known to the author.) This savings of six seconds does not mean that the user will notice a dramatic decrease in his turnaround time, but it does make available previously wasted CPU resources. The management of the computer center should make this information available to the users. WATFOR is designed to execute FORTRAN jobs that are small and that is what most of the users are submitting.

Optimizing the university computer is more difficult than optimizing the performance of a business computer center. The university must serve a wide variety of needs. The beginning programmer must be encouraged to develop good programming habits. At this institution a system that would let the user know how much of the monthly resources he used and what those resources were worth would help develop good programming habits. Too many times an instructor lets his students waste computer time discovering the solution to a problem. Careful preparation prior to initial submission of a program would save a lot of resources. Production runs of programs just to improve the format of the output may be good for a better grade, but such runs are a terrible waste if a few notes on the previous output could show where the answers were located. A program

of communication with the users and faculty should be implemented to inform users of the resources that one programming language uses compared to an alternative.

The Hardware Graph program should now be modified to produce graphs like those that appear in figures 17 and 18. The computer center at this institution has a number of plotting utility programs which could be used to fill this need for summary graphs of hardware activity. This set of summary graphs would provide the analyst with trends of the hardware activity to match the trend graphs produced by SMF Graph.

The preliminary system performance profiles presented in figures 17 and 18 indicate that the organization of data sets on selector channel 2 should be investigated. Section VII of this work detailed how to investigate the performance of a selector channel. It is important that the system performance profile be continued when measuring the performance of the selector channel. The analyst can never assume that the system is operating under the average load. The hardware monitor at this institution is ideal for such dual measurement of experiments. One event monitor can be recording the system performance profile while the second monitor records the activity of the selector channel.

It should be noted that the implementation of Quickrun has resulted in the drum storage unit being used under OS whenever CP/CMS is not operating. This is meant to alleviate the heavy paging activity, but Figures 17 and 18 clearly demonstrated that selector channel two is overworked whenever OS is operating. The use of the drum does not appear

to significantly reduce the amount of time the CPU waits for selector channel two to complete I/O.

The following sentences are a summary of the questions revealed by this work which require further research:

1. Obtain more system performance profiles. Two days is a trivial sample on which to base any decisions for system changes.
2. Present trends of performance indicators (Figures 17 and 18) to the analyst. The program SMF Graph should be modified to produce these trend graphs.
3. Perform an experiment to determine the reason for the high level of activity by Selector Channel 2.
4. Investigate reasons for the large CPU Wait only time (CPU and Selector Channel both inactive). Poor placement of data sets may be causing large disk arm seek times.
5. Determine which disk is the most active on Selector Channel 2. What would be the effect of moving the data sets from the busiest disk to the drum?
6. Determine if another 2314 Disk facility placed on Selector Channel 1 would improve performance significantly.
7. What effect would it have to use both selector channels to the same 2314 Disk facility? Can two selector channels connect to one 2314 Disk facility?

The many recent changes in operating procedures, system configuration, and task scheduling demonstrate the continuing search this institution is conducting to achieve maximum performance from the IBM 360. No computer center manager can ever be satisfied with current

performance levels. He must continue to measure system activity and improve the utilization of computer system resources.

This thesis describes the preliminary steps in optimizing the performance of a university computer system using hardware and software monitors. Although it is directed at the university computer performance problem, many of the techniques and solutions also apply to other types of computer systems, which probably have a less variable load. The use of hardware and software monitors allowed the correlation of measurement results, which lead to more meaningful indications of how to improve performance. A lot of further improvement is still possible at this installation and research, using many of the techniques in this thesis, will have to continue in order to maximize the computer's performance.

APPENDIX A

HARDWARE GRAPH PROGRAM

```
// EXEC FORTCLG
//SYSIN DD *
C
C                           HARDWARE GRAPH PROGRAM
C
C
C           THIS SUBROUTINE READS IN THE DATA FROM A SIX COUNTER MONITOR
C       RUN AND PRODUCES A SERIES OF OUTPUT BAR GRAPHS SHOWING THE TRENDS
C       OF ACTIVITY OF THE COUNTERS ON THE HARDWARE MONITOR.
C   MAXIMUM NUMBER OF INTERVALS THAT CAN BE READ INTO INPUT IS 25.


      SUBROUTINE SUB1                                                   THE00020
        REAL TITLE(9,20)                                                TES00010
        REAL INPUT(25,9)
      REAL MATRIX (27,25 )                                              THE00040
      DATA HORIZ /'****'/                                               THE00050
      DATA HOR1 /'*   '/                                                THE00060
      DATA HOR2 /'**  '/                                                THE00070
      DATA HOR3 /'*** '/                                                THE00080
       DATA HOR4 /'    '/                                               THE00090
C INT REPRESENTS THE LENGTH OF THE INTERVAL MEASURED BY MONITOR FORMAT I3
C NUMB REPRESENTS THE NUMBER OF INTERVALS BEING ANALYZED  FORMAT I2
      READ(5,911)INT,NUMB
 911  FORMAT(I3,I2)
      NUMB1=NUMB+1
C THE NINE TITLES FOR THE NINE GRAHS PRODUCED ARE READ INTO ARRAY TITLE
        DO 500 I=1,9
        READ (5,912)(TITLE(I,J),J=1,20)                                 TES00030
 912    FORMAT(20A4)                                                    TES00040
 500    CONTINUE
C THE DATA FROM THE SIX COUNTERS OF THE MONITOR IS READ INTO ARRAY INPUT
      DO 100 I=1,NUMB
      READ (5,910)(INPUT(I,J),J=1,6)                                    THE00110
910      FORMAT(6F5.2)                                                  THE00120
100   CONTINUE                                                          THE00130
C    COUNTERS 7,8,9 ARE COMPUTED FROM COUNTERS 1-6 AND PUT INTO INPUT
       DO 200 I=1,NUMB
       INPUT(I,7)= INPUT(I,1)-INPUT(I,2)                                THE00160
       INPUT(I,8)= INPUT(I,7)-INPUT(I,5)                                THE00170
       INPUT(I,9)= INPUT(I,1)-INPUT(I,5)-INPUT(I,6)                     THE00180
 200   CONTINUE                                                         THE00190
      DO 400 M=1,9                                                      THE00140
C  THIS INITIALIZES THE MATRIX GRAPH TO A BLANK PAGE WITH SCALES AND
C     BORDERS INSERTED.
      CALL PIC(MATRIX)                                                  THE00200
C THE LENGTH OF EACH BAR ON THE GRAPH IS COMPUTED HERE.
      DO 300 J= 2,NUMB1
      IRESUL = IFIX( INPUT((J-1),M ))                                   THE00220
```

```
      IRESUL=IRESUL / 4                                                 THE00230
      IREM = IFIX( INPUT( (J-1),M)) - (IRESUL * 4)                      THE00240
      DO 301 K= 1, IRESUL                                               THE00250
      MATRIX (J,K) = HORIZ                                              THE00260
301   CONTINUE                                                          THE00270
      TEMP=HOR4                                                         THE00280
      IF(IREM.EQ.1)TEMP=HOR1                                            THE00290
      IF(IREM.EQ.2)TEMP=HOR2                                            THE00300
      IF(IREM.EQ.3)TEMP=HOR3                                            THE00310
      MATRIX(J,(IRESUL + 1 ))= TEMP                                     THE00320
300   CONTINUE                                                          THE00330
C  OUTPUT OF THE GRAPH JJ IS DONE HERE.  JJ=1,9.
        WRITE(6,920)(TITLE(M,J),J=1,20)                                 TES00070
 920    FORMAT('1',25X,20A4,//)                                         TES00080
      WRITE(6,921)                                                      THE00360
 921  FORMAT('0',8X,'TIME',13X,'    4   8  12  16  20  24  28  32  36  40
     2  44  48  52  56  60  64  68  72  76  80  84  88  92  96 100')    THE0038
      WRITE(6,923)(MATRIX(1,N),N=1,25)                                  THE00390
      I=INT
      DO 6 JJ= 2,NUMB1
      WRITE(6,922)I,INPUT((JJ-1),M),(MATRIX(JJ,N),N=1,25)
 922  FORMAT('0',8X,I4,8X,F5.2,25A4,/)
      I=I+INT
6     CONTINUE                                                          THE00450
      WRITE(6,923)(MATRIX(27,N),N=1,25)                                 THE00460
923   FORMAT('0',25X,25A4)
       WRITE(6,921)                                                     THE00480
 400  CONTINUE                                                          THE00490
      RETURN                                                            THE00500
      END                                                               THE00510


      SUBROUTINE PIC (MATRIX)                                           THE00520
      REAL MATRIX(27,25)                                                THE00530
      DATA BAR /'****'/                                                 THE00540
       DATA VERT /'    '/
      DATA HORIZ /'----'/                                               THE00570
      DO 1 I=1,25                                                       THE00580
      MATRIX(1,I)=HORIZ                                                 THE00590
      MATRIX(27,I)=HORIZ                                                THE00600
1     CONTINUE                                                          THE00610
      DO 3 I=2,26                                                       THE00620
      DO 2 J=1,25                                                       THE00630
      MATRIX(I,J)=VERT                                                  THE00640
2     CONTINUE                                                          THE00650
3     CONTINUE                                                          THE00660
      RETURN                                                            THE00670
      END                                                               THE00680
```

```
        REAL TITLE(9,20)                                                        THE00700
        REAL INPUT(25,9)                                                        THE00710
      REAL MATRIX (27,25 )                                                      THE00720
      CALL SUB1
      STOP
      END                                                                       THE00730
//GO.SYSIN DD *
03011
                           MACHINE NOT MANUAL
                                WAIT
                         CPU WAIT AND CHANNEL 2 BUSY
                     CPU ACTIVE AND CHANNEL 2 BUSY
                                  CPU ONLY
                              CHANNEL 2 BUSY
                         COMPUTE
                             CHANNEL 2 ONLY
                                WAIT ONLY
99.9957.4738.3222.7619.7761.08
97.1558.0528.7823.0318.9251.81
99.9971.7348.3114.4913.7862.80
99.9970.1547.4014.2415.6161.64
91.3276.6840.2310.4412.4550.67
99.9973.4746.3111.4715.0657.78
98.9264.8227.7812.4521.6440.23
99.4486.4329.2205.7007.8834.92
99.9978.7649.0209.8811.3658.90
99.9938.5323.8920.5140.9644.40
99.9955.4327.4123.4021.1750.81
```

APPENDIX B

SMF GRAPH PROGRAM

```
//  EXEC  FORTCLG
//SYSIN DD *
C
C                          SMF GRAPH PROGRAM
C
C
C TTIME IS THE TOTAL SYSTEM TIME USED BY ALL TYPE STEPS
C  STEPTM IS THE SYSTEM TIME USED BY ONE STEP
C  KK IS THE INTERVAL NUMBER COREESPONDING TO MONITOR TAPE
C  INT IS INTERVAL SIZE IN SECONDS  FORMAT I5
C  NUM IS THE NUMBER OF INTERVALS ON THE MONITOR TAPE   FORMAT I2
C  ITIME IS THE INITIAL TIME OF MEASUREMENT    EXAMPLE  1053
C
      INTEGER Q1/'QUIC'/,Q2/'KRUN'/
      INTEGER QUKSTP(100)/100*0/,QUIKTM(100)/100*0/,QUKCPU(100)/100*1/
      INTEGER  CPU
      REAL TOTAL/0.0/
      INTEGER  FOGCPU(100)/100*1/,LINCPU(100)/100*1/,GOCPU(100)/100*1/
      INTEGER WATCPU(100)/100*1/,WCTCPU(100)/100*1/,ALGCPU(100)/100*1/
      INTEGER FOHCPU(100)/100*1/,GPSCPU(100)/100*1/
      INTEGER WATSTP(100)/100*0/,ALGSTP(100)/100*0/,ALTIME(100)/100*0/
      INTEGER FORTG(100)/100*0/,FORGTM(100)/100*0/,LINK(100)/100*0/
      INTEGER          STEPTM /0/, KK /1/
      INTEGER WATCTP(100)/100*0/,WCTIME(100)/100*0/
      INTEGER TTIME(100)/100*1/,NSTEPS(100)/100*1/,WATIME(100)/100*0/
      REAL INPUT(25,9)
      INTEGER TITLE(9,20)
      INTEGER*2 SWITCH /0/
      INTEGER LINKTM(100)/100*0/,FORTH(100)/100*0/,FORHTM(100)/100*0/
      INTEGERGPSS(100)/100*0/,GPSSTM(100)/100*0/,FORGO(100)/100*0/
      INTEGER  GOTIME(100)/100*0/
C
C TI THROUGH T9 REPRESENT THE NINE TYPES OF JOB STEPS THAT STATISTICS
C  WILL BE GATHERED ON.  FORTRAN COMPILATIONS, GPSS, AND ALGOL ARE
C  THREE OF THE TYPES WHICH ARE TABULATED BY THIS VERSION.  SIMPLY
C  CHANGING THE INITIALIZATION OF T1 TO THE PGM IDENTIFIER OF THE
C  TYPE YOU ARE INTERESTED IN ALLOWS STATISTICS TO BE GATHERED ON IT.
C
      INTEGER T1 /'WAT5'/,T2 /'TIME'/,T3 /'ALGO'/,T4 /'IEYF'/
      INTEGER  T5 /'IEWL'/,T6 /'PGM='/,T7 /'IEKA'/, T8 /'DAGO'/
      INTEGER T9/'QUIC'/
      INTEGER*2 SUB,REST,REST2
      INTEGER START
      INTEGER   RNAME,RNAM1,SNAME,SNAM1
      INTEGER DATE
      INTEGER SMF(100),IC,JC,DATE,BINDAT,TIME
       INTEGER*2 SMF2(201),TYPE,STPREC/4/,SWTCHX/0/,FIRSTX/0/
      EQUIVALENCE (SMF(1),SMF2(2))
```

```
C
CERRSET SUPPRESSES THE COMPILER ERROR MESSAGES WHEN THE END OF EACH SMF
C  RECORD IS READ.
C
      CALL ERRSET(213,300,-1,1,0,0)
      READ(5,40)INT,NUM,ITIME
 40   FORMAT(I5,I2,I4)
      WRITE(6,41)ITIME,INT
 41   FORMAT('0**** MEASUREMENTS BEGAN AT ',I4,' WITH INTERVALS',I5,
     2' SECONDS LONG ***')
      ITEMP = ITIME / 100
      IREM = ITIME - (ITEMP * 100)
C
C   INIT IS INTITAL MEASUREMENT TIME IN 100THS OF SECONDS
C
      INIT= ((ITEMP * 3600) + (IREM * 60)) * 100
C
C  INT IS THE INTERVAL SIZE IN 100THS OF SECONDS
C
      INT = INT * 100
C
C  NEXT IS THE START OF THE SECOND INTERVAL
C
      NEXT = INIT + INT
      WRITE(6,111)
 111  FORMAT('OJOBNAME',5X,'START',9X,'STOP', 9X,'DURATION',5X,
    2 'PROGNAME'4X,'TYPE',5X,'CPU SECONDS')
C
C  THIS SECTION OF CODE SELECTS THE CURRENT OPEN SMF DISK FILE TO BE READ
C  THERE ARE TWO FILES HERE AT NPS, SYS1.MANX AND SYS1.MANY.
C
      WRITE(6,1000)
 1000 FORMAT(1X,'TRYING TO READ FROM SYS1.MANX')
  1   IF(SWTCHX.EQ.1)GO TO 5
      READ(10,END=3,ERR=4)SMF2
  4   CONTINUE
      FIRSTX=1
      GO TO 9
  3   IF(FIRSTX.NE.0)GO TO 100
      SWTCHX=1
      WRITE(6,1001)
 1001 FORMAT(1X,'READING FROM SYS1.MANY')
  5   READ(11,END=100,ERR=6)SMF2
   6  CONTINUE
C
C  CHMOVE EXTRACTS BYTE 2 FROM SMF2(1) AND PUTS IT INTO BYTE 2 OF TYPE
C
  9   TYPE=0
```

```
      CALL CHMOVE(SMF2(1),2,TYPE,2)
C
C  HERE WE ARE LOOKING FOR JOB STEP TERMINATION RECORDS.  THEY ARE TYPE
C  OF FOUR.  THERE ARE FOURTEEN DIFFERENT TYPES OF SMF RECORDS.
C
      IF(TYPE.EQ.128)GO TO 1100
      IF(TYPE.NE.4)GO TO 1
      DO 10 J=1,4
   10 CALL CHMOVE(SMF2(2),J,TIME,J)
C
C  THIS STEPS THROUGH SMF RECORDS FROM PREVIOUS DAY
C
      IF(TIME.GT. NEXT .AND. SWITCH .EQ. 0 ) GO TO 1
C
C  SNAME REPRESENTS THE SYSTEM PROGRAM EXECUTED BY THIS JOB STEP.
C
      DO 12 K=1,4
   12 CALL CHMOVE(SMF2(28),K,SNAME,K)
      DO 13 K=1,4
  13  CALL CHMOVE(SMF2(30),K,SNAM1,K)
C
C  RNAME IS THE USER IDENTIFICATION FOR HIS JOB.
C
      DO 77 K=1,4
   77 CALL CHMOVE(SMF2( 8),K,RNAME,K)
      DO 78 K=1,4
  78  CALL CHMOVE(SMF2(10),K,RNAM1,K)
      SUB=0
      REST=0
      REST2=0
C
C  START IS THE JOB STEP INITIATION TIME OF A JOB STEP.
C
      START=0
      CALL CHMOVE(SMF2(20),2,SUB,2)
      CALL CHMOVE(SMF2(21),1,REST,1)
      CALL CHMOVE(SMF2(21),2,REST,2)
      CALL CHMOVE(SMF2(22),1,REST2,2)
      START=(REST * 256)+REST2 + (SUB * (2**24))
C
C  STEPTM IS JOB TERMINATION TIME MINUS JOB INITIATION TIME.
C
      STEPTM=TIME - START
C
C  CPU IS THE CPU SECONDS USED BY THE JOB STEP DURING STEPTM.
C
      CPU=0
      M=0
```

```
      DO 717 I=1,2
      I2=I+2
 717  CALL CHMOVE(SMF2(52),I,M,I2)
      DO 718 J=2,4
      N=M+J
 718  CALL CHMOVE(SMF2(52),N,CPU,J)
C
C  SMF JOB STEP TERMINATION RECORDS ARE PRINTED HERE.
C
 1105 WRITE(6,110)RNAME,RNAM1,START,TIME,STEPTM,SNAME,SNAM1,TYPE,CPU
 110  FORMAT(' ',A4,A4,5X,I8,5X,I8,5X,I8,5X,A4,A4,5X,I3,5X,I8)
C
C  JOB STEP NOT IN INTERVALL KK
C
      IF(TIME.LT.INIT)GO TO 1
C
C  PRINT A LINE OF * * * TO INDICATE BEGINNING OF FIRST INTERVAL
C
      IF(SWITCH.EQ.1)GO TO 33
      SWITCH=1
      WRITE(6,62)
      INT2=INT/100
      WRITE(6,41)ITIME,INT2
 33   CONTINUE
C
C  END OF INTERVAL KK SUM UP STATISTICS FOR INTERVAL
C
  55  IF(TIME.GT.NEXT)GO TO 60
C
C  DETERMINE WHICH TYPE OF PROGRAM WAS EXECUTED BY STEP AND RECORD STATISTICS
C
      IF(SNAME.EQ.T1)GO TO 200
      IF(SNAME.EQ.T2)GO TO 300
      IF(SNAME.EQ.T3)GO TO 400
      IF(SNAME.EQ.T4)GO TO 500
      IF(SNAME.EQ.T5)GO TO 600
      IF(SNAME.EQ.T6)GO TO 660
      IF(SNAME.EQ.T7)GO TO 700
      IF(SNAME.EQ.T8)GO TO 800
      IF(SNAME.EQ.T9)GO TO 900
C
C   JOB STEP IS NOT OF TYPE WE ARE INTERESTED IN
C  ADD STEP TIME  TO TOTAL  FOR INTERVALL KK
C
 101  NSTEPS(KK) =NSTEPS(KK) + 1
      TTIME(KK) = TTIME(KK) + STEPTM
      GO TO 1
C
```

```
C  WATFOR STATISTICS ARE GATHERED AT 200 AND 300
C
 200   WATSTP(KK)= WATSTP(KK)+1
       WATIME(KK) = WATIME(KK) + STEPTM
       WATCPU(KK)=WATCPU(KK)+CPU
       GO TO 101
C
C  WATFORC STATS ARE GATHERED AT 300
C
 300   WATCTP(KK)=WATCTP(KK)+1
       WCTIME(KK)=WCTIME(KK)+1
       WCTCPU(KK)=WCTCPU(KK)+CPU
       GO TO 101
C
C ALGOL STEP INFORMATION
C
 400   ALGSTP(KK) = ALGSTP(KK) + 1
       ALTIME(KK) = ALTIME(KK) +STEPTM
       ALGCPU(KK)=ALGCPU(KK)+CPU
       GO TO 101
C
C  FORTRAN G COMPILATION STATS
C
 500   FORTG(KK) = FORTG(KK) + 1
       FORGTM(KK)=FORGTM(KK) + STEPTM
       FOGCPU(KK)=FOGCPU(KK)+CPU
       GO TO 101
C
C   FORTRAN LINK STEPS GATHERED HERE
C
 600   LINK(KK) =LINK(KK) + 1
       LINKTM(KK)=LINKTM(KK)+STEPTM
       LINCPU(KK)=LINCPU(KK)+CPU
       GO TO 101
C
C  FORTRAN GO STEPS ARE GATHERED HERE
C
 660   FORGO(KK) = FORGO(KK) +1
       GOTIME(KK)=GOTIME(KK) + STEPTM
        GOCPU(KK)= GOCPU(KK)+CPU
       GO TO 101
C
C  FORTRAN H COMPILATION ARE GATHERED
C
 700   FORTH(KK) = FORTH(KK) + 1
       FORHTM(KK) =FORHTM(KK) + STEPTM
       FOHCPU(KK)=FOHCPU(KK)+CPU
       GO TO 101
```

```
C
C   GPSS STATISTICS ARE GATHERED
C
 800  GPSS(KK) = GPSS(KK) + 1
      GPSSTM(KK) =GPSSTM(KK)+STEPTM
      GPSCPU(KK)=GPSCPU(KK)+CPU
      GO TO 101
C
C  QUICKRUN STATISTICS ARE GATHERED HERE
C
 900  QUKSTP(KK)=QUKSTP(KK)+1
      QUIKTM(KK)=QUIKTM(KK)+STEPTM
      QUKCPU(KK)=QUKCPU(KK)+CPU
      GO TO 101
C
C   THIS SECTION BREAKSDOWN THE QUICKRUN SMF RECORD TYPE 128
C
 1100 DO 1104 I=1,2
 1104 CALL CHMOVE(SMF2(2),I,RNAME,I)
      DO 1107 I=3,4
 1107 CALL CHMOVE(SMF2(2),I,RNAM1,I)
      SNAME=Q1
      SNAM1=Q2
      TIME=0
      DO 1101 I=1,4
 1101 CALL CHMOVE(SMF2(22),I,TIME,I)
      IF(TIME.GT.NEXT.AND.SWITCH.EQ.0)GO TO 1
      START=0
      DO 1102 I=1,4
 1102 CALL CHMOVE(SMF2(6),I,START,I)
      CPU=0
      DO 1103 I=1,4
 1103 CALL CHMOVE(SMF2(20),I,CPU,I)
      STEPTM=TIME-START
      GO TO 1105
C
C  END OF INTERVAL STATISTICS ARE RECORDED IN THIS SECTION.
C
 60   NEXT=NEXT+INT
C
C INT IS THE END OF NEXT INTERVAL
C  INIT IS THE START OF NEXT INTERVAL
C
      INIT = INIT + INT
      K2=KK+1
      WRITE(6,61)K2
 61   FORMAT('0',70X,'ABOVE JOB STEP STARTS INTERVAL  ',I2)
      WRITE(6,62)
```

```
62    FORMAT('0* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ',
     2'* * * * * * * * * * * * * * * * * * * * * * * ')
      WRITE(6,111)
      KK=KK+1
      IF(KK.GT.NUM)GO TO 100
      GO TO 55
100   WRITE(6,11)
11    FORMAT('0*********FINISHED  SMF  *******')
      K1=KK-1
C
C  THIS SECTION PRODUCES THE STEP GRAPHES SHOWING NUMBER OF STEPS BY
C  TYPE THAT WERE EXECUTED DURING EACH INTERVAL.
C
      DO 701 J=1,K1
      INPUT(J,1)=(FLOAT(WATSTP(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,2)=(FLOAT(WATCTP(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,3)=(FLOAT(ALGSTP(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,4)=(FLOAT( FORTG(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,5)=(FLOAT(  LINK(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,6)=(FLOAT( FORGO(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,7)=(FLOAT( FORTH(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,8)=(FLOAT(  GPSS(J)) / FLOAT(NSTEPS(J)))*100
      INPUT(J,9)=(FLOAT(QUKSTP(J)) / FLOAT(NSTEPS(J)))*100
701   CONTINUE
      CALL SUB1(INPUT,K1)
C
C  THIS SECTION PRODUCES GRAPHES WHICH SHOW HOW MUCH OF THE TOTAL TIME
C  RECORDED IN EACH INTERVAL WENT TO THE PROCESSING OF EACH JOB STEP TYPE.
C
      DO 702 J=1,K1
      INPUT(J,1)=(FLOAT(WATIME(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,2)=(FLOAT(WCTIME(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,3)=(FLOAT(ALTIME(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,4)=(FLOAT(FORGTM(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,5)=(FLOAT(LINKTM(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,6)=(FLOAT(GOTIME(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,7)=(FLOAT(FORHTM(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,8)=(FLOAT(GPSSTM(J)) / FLOAT( TTIME(J)))*100
      INPUT(J,9)=(FLOAT(QUIKTM(J)) / FLOAT(TTIME(J)))*100
      TOTAL=TOTAL+TTIME(J)
702   CONTINUE
      CALL SUB1(INPUT,K1)
C
C AVERAGE JOB STEPS ACTIVE IN INTERVALL KK IS AVE.
C
      DO 708 K=1,K1
      AVE=FLOAT(TTIME(K))/FLOAT(INT)
      WRITE(6,704)AVE,K
```

```
 704  FORMAT('OAVERAGE NUMBER JOB STEPS WAS',F10.5,'DURING INTERVAL',I2)
 708  CONTINUE
C
C TAVE  IS THE AVERAGE NUMBER OF STEPS TERMINATED OFER ENTIRE PERIOD
C
      INTLNG=INT*K1
      TAVE=TOTAL/INTLNG
      WRITE(6,705)TAVE
 705  FORMAT('OAVERAGE NUMBER JOBS ACTIVE OVER ENTIRE PERIOD ',F10.5)
C
C  THIS PRODUCES TABLES WHICH SHOW TYPE STEPS EXECUTED IN INTERVAL I.
C  CORE USE IS THE RATIO OF TIME IN MACHINE TO CPU SECONDS USED
C
      DO 801 I=1,K1
      WRITE(6,62)
      WRITE(6,802)I
 802  FORMAT('OJOB STEPS RECORDED IN INTERVAL'8I2)
      RI=FLOAT(WATIME(I))/FLOAT(WATCPU(I))
      WRITE(6,803)WATSTP(I),RI
 803  FORMAT('OWATFOR JOB STEPS EQUAL',I2,20X,'CORE USE ', F10.5)
      RI=FLOAT(WCTIME(I))/FLOAT(WCTCPU(I))
      WRITE(6,804) WATCTP(I),RI
 804  FORMAT('OWATFOR C JOB STEPS EQUAL ',I2,20X,'CORE USE ',F10.5)
      RI=FLOAT(ALTIME(I))/FLOAT(ALGCPU(I))
      WRITE(6,805)ALGSTP(I),RI
 805  FORMAT('OALGOL JOB STEPS EQUAL ',I2,20X,'CORE USE ',F10.5)
      RI=FLOAT(FORGTM(I))/FLOAT(FOGCPU(I))
      WRITE(6,806)FORTG(I),RI
 806  FORMAT('OFORTRAN G COMPILE STEPS EQUAL',I2,20X,'CORE USE ',F10.5)
      RI=FLOAT(LINKTM(I))/FLOAT(LINCPU(I))
      WRITE(6,807)LINK(I),RI
 807  FORMAT('OFORTRAN LINK STEPS EQUAL ',I2,20X,'CORE USE ',F10.5)
      RI=FLOAT(GOTIME(I))/FLOAT(GOCPU (I))
      WRITE(6,808)FORGO(I),RI
 808  FORMAT('OFORTRAN GO STEPS EQUAL ',I2,20X,'CORE USE ',F10.5)
      RI=FLOAT(FORHTM(I))/FLOAT(FOHCPU(I))
      WRITE(6,809)FORTH(I),RI
 809  FORMAT('OFORTRAN H COMPILE STEPS EQUAL ',I2,20X,'CORE USE',F10.5)
      RI=FLOAT(GPSSTM(I))/FLOAT(GPSCPU(I))
      WRITE(6,810)GPSS(I),RI
 810  FORMAT('OGPSS STEPS EQUAL ',I2,20X,'CORE USE',F10.5)
      RI=FLOAT(QUIKTM(I)) / FLOAT(QUKCPU(I))
      WRITE(6,811)QUKSTP(I),RI
 811  FORMAT('OQUICKRUN STEPS EQUAL ',I2,20X,'CORE USE',F10.5)
C
      WRITE(6,62)
 801  CONTINUE
      STOP
```

```
      END

      SUBROUTINE CHMOVE(A,I,B,J)
      LOGICAL*1  A(50),B(10)
      B(J)=A(I)
      RETURN
      END

      SUBROUTINE SUB1(INPUT,KK)                                          THE00080
C         THIS SUBROUTINE READS IN THE DATA FROM A SIX COUNTER MONITOR   THE00040
C     RUN AND PRODUCES A SERIES OF OUTPUT BAR GRAPHS SHOWING THE TRENDS  THE00050
C     OF ACTIVITY OF THE COUNTERS ON THE HARDWARE MONITOR.               THE00060
C   MAXIMUM NUMBER OF INTERVALS THAT CAN BE READ INTO INPUT IS 25.       THE00070
        REAL TITLE(9,20)                                                 THE00090
      REAL MATRIX (27,25 ) ,INPUT(25,9)                                  THE00110
      DATA HORIZ /'****'/                                                THE00120
      DATA HOR1 /'*   '/                                                 THE00130
      DATA HOR2 /'**  '/                                                 THE00140
      DATA HOR3 /'*** '/                                                 THE00150
       DATA HOR4 /'    '/                                                THE00160
C INT REPRESENTS THE LENGTH OF THE INTERVAL MEASURED BY MONITOR FORMAT ITHE00170
C NUMB REPRESENTS THE NUMBER OF INTERVALS BEING ANALYZED  FORMAT I2      THE00180
      NUMB=KK
      NUMB1=NUMB+1                                                       THE00210
C THE NINE TITLES FOR THE NINE GRAHS PRODUCED ARE READ INTO ARRAY TITLE THE00220
        DO 500 I=1,9                                                     THE00230
        READ (5,912)(TITLE(I,J),J=1,20)                                  THE00240
 912    FORMAT(20A4)                                                     THE00250
 500    CONTINUE                                                         THE00260
      DO 400 M=1,9                                                       THE00380
C   THIS INITIALIZES THE MATRIX GRAPH TO A BLANK PAGE WITH SCALES AND    THE00390
C     BORDERS INSERTED.                                                  THE00400
      CALL PIC(MATRIX)                                                   THE00410
C THE LENGTH OF EACH BAR ON THE GRAPH IS COMPUTED HERE.                  THE00420
      DO 300 J= 2,NUMB1                                                  THE00430
      IRESUL = IFIX( INPUT((J-1),M ))                                    THE00440
       IRESUL=IRESUL / 4                                                 THE00450
      IREM = IFIX( INPUT( (J-1),M)) - (IRESUL * 4)                       THE00460
      DO 301 K= 1, IRESUL                                                THE00470
      MATRIX (J,K) = HORIZ                                               THE00480
301   CONTINUE                                                           THE00490
      TEMP=HOR4                                                          THE00500
      IF(IREM.EQ.1)TEMP=HOR1                                             THE00510
      IF(IREM.EQ.2)TEMP=HOR2                                             THE00520
      IF(IREM.EQ.3)TEMP=HOR3                                             THE00530
      MATRIX(J,(IRESUL + 1 ))= TEMP                                      THE00540
```

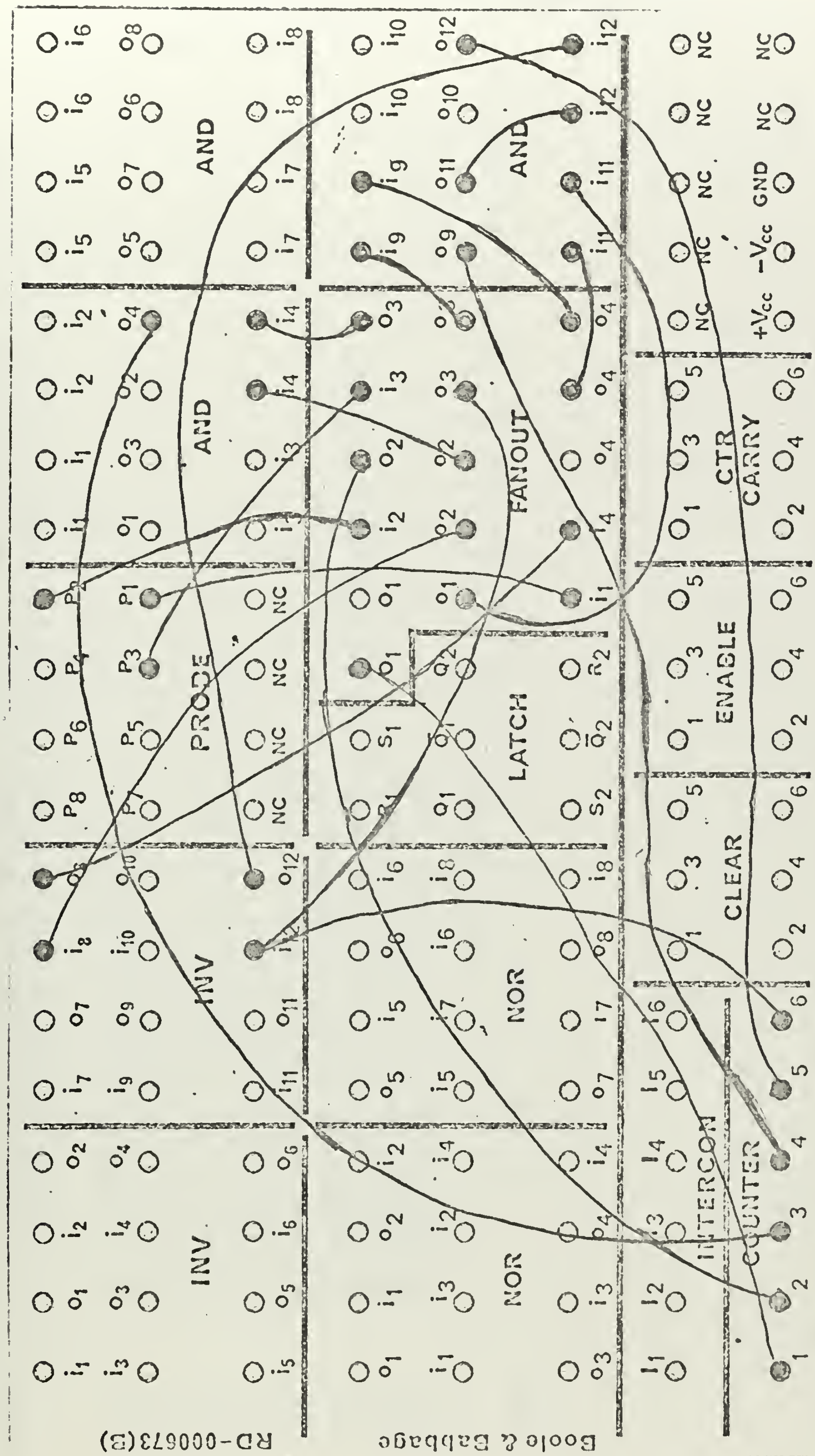
```
300    CONTINUE
C   OUTPUT OF THE GRAPH JJ IS DONE HERE.   JJ=1,9.
          WRITE(6,920)(TITLE(M,J),J=1,20)
 920      FORMAT('1',25X,20A4,//)
       WRITE(6,921)
 921   FORMAT('0','INTERVAL',9X,'    4    8   12   16   20   24   28   32   36   40
     2  44   48   52   56   60   64   68   72   76   80   84   88   92   96  100')
       WRITE(6,923)(MATRIX(1,N),N=1,25)
       DO 6  JJ= 2,NUMB1
       J=JJ-1
       WRITE(6,922)J,INPUT((JJ-1),M),(MATRIX(JJ,N),N=1,25)
 922   FORMAT('0',2X,I2,8X,F5.2,25A4,/)
6      CONTINUE
       WRITE(6,923)(MATRIX(27,N),N=1,25)
923    FORMAT('0',17X,25A4)
        WRITE(6,921)
 400   CONTINUE
       RETURN
       END


       SUBROUTINE PIC  (MATRIX)
       REAL MATRIX(27,25)
       DATA BAR /'****'/
        DATA VERT /'    '/
       DATA HORIZ /'----'/
       DO 1  I=1,25
       MATRIX(1,I)=HORIZ
       MATRIX(27,I)=HORIZ
1      CONTINUE
       DO 3  I=2,26
       DO 2  J=1,25
       MATRIX(I,J)=VERT
2      CONTINUE
3      CONTINUE
       RETURN
       END
//GO.FT06F001  DD DCB=(RECFM=FBA,LRECL=133,BLKSIZE=133),
//          SPACE=(CYL,(2,1))
//GO.FT10F001   DD   DSNAME=SYS1.MANX,DISP=SHR
//GO.FT11F001   DD   DSNAME=SYS1.MANY,DISP=SHR
//GO.SYSIN DD *
01800240800
                        PERCENT    WATFOR STEPS
                        PERCENT WATFORC STEPS
                    PERCENT ALGOL STEPS
                        PERCENT FORTG COMPILATION STEPS
                      PERCENT  FORTRAN LINK STEPS
```

```
 PERCENT FORTRAN  GO STEPS
PERCENT FORTRAN H COMPILE STEPS
   PERCENT  GPSS STEPS
    PERCENT QUICKRUN STEPS
PERCENT  WATFOR TIME
    PERCENT  WATFORC  TIME
   PERCENT    ALGOL TIME
    PERCENT   FORTRAN G TIME
   PERCENT  FORT LINK TIME
    PERCENT FORT GO TIME
    PERCENT FORTRAN H COMPILE TIME
     PERCENT GPSS TIME
   PERCENT QUICKRUN TIME
```

APPENDIX C

SYSTEM PERFORMANCE PROFILE LOGIC PLUGBOARD

1. Manual Mode 0 E-E2B4 B08 CPU 2. Wait + E-C2J6 B07 CPU 3. Chan 2 Busy + B-A3L5 B10 2860



1. $\overline{\text{Manual Mode}}$
2. Wait
3. Chan 2 Busy & Wait (CPU waiting on I/O)
4. Chan 2 Busy & $\overline{\text{Wait}}$ (Overlap)
5. Manual Mode & $\overline{\text{Wait}}$ & $\overline{\text{Chan 2 Busy}}$ (CPU only)
6. Chan 2 Busy

APPENDIX D

CPU UTILIZATION (EXPERIMENT 4)

GRAPHS FROM HARDWARE GRAPH

COMPUTE

TIME	COMPUTE
1100	26.73
1130	23.27
1200	16.38
1230	18.41
1300	27.97
1330	17.30
1400	27.12
1430	25.09
1500	56.88
1530	26.06
1600	28.49

Scale: 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68

MACHINE NOT MANUAL

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                ________________________________________________________________________

1100       99.99************************************************************************
1130       99.99************************************************************************
1200       98.42************************************************************************
1230       99.99************************************************************************
1300       99.99************************************************************************
1330       97.00************************************************************************
1400       99.99************************************************************************
1430       99.97************************************************************************
1500       99.99************************************************************************
1530       99.99************************************************************************
1600       99.74************************************************************************
                ________________________________________________________________________

TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

CHANNEL 2 BUSY

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                   ______________________________________________________________________

1100     54.80*******************************************************
1130     55.18********************************************************
1200     40.65*****************************************
1230     45.08**********************************************
1300     51.24****************************************************
1330     50.34***************************************************
1400     47.19************************************************
1430     48.65*************************************************
1500     44.86*********************************************
1530     49.62**************************************************
1600     59.12************************************************************
                   ______________________________________________________________________

TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

CPU ACTIVE AND CHANNEL 2 BUSY

```
TIME                 4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                  ______________________________________________________________________

1100        11.28***********
1130        11.08***********
1200         7.89*******
1230         7.50*******
1300        12.01************
1330         8.83********
1400        10.16**********
1430        10.76**********
1500        22.21**********************
1530        11.36***********
1600        14.27**************
                  ______________________________________________________________________

TIME                 4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

CPU WAIT AND CHANNEL 2 BUSY

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                 ------------------------------------------------------------------------

1100       43.52*******************************************
1130       44.10********************************************
1200       32.76********************************
1230       37.57*************************************
1300       39.23***************************************
1330       41.51*****************************************
1400       37.03*************************************
1430       38.89**************************************
1500       22.65**********************
1530       38.26**************************************
1600       44.85********************************************

                 ------------------------------------------------------------------------
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

CPU ONLY

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                 ________________________________________________________________________

1100      15.46***************
1130      12.20************
1200      10.07**********
1230      10.97**********
1300      15.92***************
1330      11.27***********
1400      17.39*****************
1430      14.34**************
1500      34.68**********************************
1530      14.71**************
1600      14.48**************
                 ________________________________________________________________________

TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

PERCENTAGE FORTG COMPILATION STEPS

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64
                 ----------------------------------------------------------------

1100       10.94**********
1130       17.74*****************
1200       14.29**************
1230        9.26*********
1330       11.48***********
1400       13.56*************
1430       16.67****************
1500       15.94***************
1530       12.36************
1600       17.72*****************

                 ----------------------------------------------------------------
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64
```

PERCENTAGE WATFOR STEPS

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64
                  ______________________________________________________________

1100      0.0
1130      1.61*
1200      0.0
1230      1.85*
1330      0.0
1400      8.47********
1430      1.28*
1500      1.45*
1530      0.0
1600      0.0

                  ______________________________________________________________

TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64
```

PERCENTAGE WATFORC STEPS

TIME		4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64
1100	0.0																
1130	0.0																
1200	0.0																
1230	0.0																
1330	0.0																
1400	1.69*																
1430	0.0																
1500	1.45*																
1530	0.0																
1600	0.0																
TIME		4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64

PERCENTAGE ALGOL STEPS

TIME		4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64
1100	0.0																
1130	1.61*																
1200	1.59*																
1230	0.0																
1330	0.0																
1400	0.0																
1430	1.28*																
1500	0.0																
1530	0.0																
1600	0.0																
TIME		4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64

PERCENTAGE FORTRAN LINK STEPS

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                 ______________________________________________________________________

1100     12.50************
1130     22.58**********************
1200     22.22**********************
1230     14.81**************
1330     16.39****************
1400     13.56*************
1430     23.08***********************
1500     17.39*****************
1530     15.73***************
1600     20.25********************
                 ______________________________________________________________________

TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

PERCENTAGE FORTRAN GO STEPS

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                 ------------------------------------------------------------------------

1100      12.50************
1130      19.35*******************
1200      19.05*******************
1230      14.81**************
1330      18.03******************
1400      13.56*************
1430      21.79*********************
1500      18.84******************
1530      13.48*************
1600      22.78**********************

                 ------------------------------------------------------------------------
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

PERCENTAGE FORTRAN H COMPILE STEPS

TIME		4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72
1100	0.00																		
1130	0.00																		
1200	0.00																		
1230	0.00																		
1300	0.00																		
1330	0.00																		
1400	0.00																		
1430	0.00																		
1500	0.00																		
1530	0.00																		
1600	0.00																		
TIME		4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72

PERCENTAGE GPSS STEPS

```
TIME              4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                  ---------------------------------------------------------------------
1100      7.81*******
1130      0.00
1200      0.00
1230      0.00
1330      0.00
1400     10.17**********
1430      0.00
1500      0.00
1530      1.12*
1600      0.00
                  ---------------------------------------------------------------------
TIME              4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

PERCENTAGE QUICKRUN STEPS

```
TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
                ______________________________________________________________________

1100      31.25*******************************
1130      .24.19************************
1200      30.16******************************
1230      33.33*********************************
1330      34.43**********************************
1400      30.51******************************
1430      25.64*************************
1500      34.78**********************************
1530      39.33***************************************
1600      30.38******************************
                ______________________________________________________________________

TIME                4   8  12  16  20  24  28  32  36  40  44  48  52  56  60  64  68  72
```

TABLE FROM SMF GRAPH

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 2.9155 DURING INTERVAL 1100

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 2.2967 DURING INTERVAL 1130

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 4.0829 DURING INTERVAL 1200

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 2.1191 DURING INTERVAL 1230

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 2.1592 DURING INTERVAL 1330

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 3.5795 DURING INTERVAL 1400

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 2.6053 DURING INTERVAL 1430

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 3.6347 DURING INTERVAL 1500

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 2.5727 DURING INTERVAL 1530

AVERAGE NUMBER OF JOB STEPS ACTIVE WAS 1.4671 DURING INTERVAL 1600

LIST OF REFERENCES

1. Warner, C. D., "Monitoring: A Key to Cost Efficiency," Datamation, 1. 40-49, 1 January 1971.

2. Deese, D. R., Hardware Monitors in the User Environment, paper presented at SHARE XXXVI, Los Angeles, California, 11 March 1971.

3. Hart, L. E., "The User's Guide to Evaluation Products," Datamation, 1. 32-35, 15 December 1970.

4. Bonner, A. J., "Using System Monitor Output to Improve Performance," IBM Systems Journal, Vol. 4, p. 290-297, 1969.

5. International Business Machines Corporation, IBM System 360 Operating System, Planning for System Management Facilities, Second Edition, December 1969.

6. International Business Machines Corporation, IBM System 360 Operating System, System Programmer's Guide, 1969.

7. Boole & Babbage, Incorporated, Measurement Engine Applications Handbook, 1971.

8. Cockrum, J. S. and Crockett, E.D., Interpreting the Results of a Hardware Monitor, paper obtained from Computer Synectics Incorporated, Santa Clara, California, 1971.

9. Carlson, G., A User's View of Hardware Performance Monitors or How to Get More Computer for Your Dollar, paper published in Proceedings IFIP Congress, 23-28 August 1971.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assistant Professor G. H. Syms, Code 53Zz Department of Mathematics Naval Postgraduate School Monterey, California 93940	2
4.	Instructor Gary A. Kildall, Code 53Kd Department of Mathematics Naval Postgraduate School Monterey, California	1
5.	LT Robert R. Hanke, USN 339 Brice Avenue Mundelein, Illinois 66060	1

Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

REPORT TITLE

Preliminary Steps in Optimizing University Computer Performance Using Hardware and Software Monitors

DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; December 1971

AUTHOR(S) (First name, middle initial, last name)

Robert Richard Hanke

REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1971	103	9

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

Too much money is being spent on new computer systems without any idea of what the new systems can do. The large expenditures for computer hardware necessitates obtaining the maximum performance for every dollar spent, in order for the computer system to be cost effective.

This research effort explores the process of selecting, implementing, and using a hardware monitor to measure the performance of a university computer system. Information about the work being performed by the computer system was obtained without the use of a special software monitor, instead the System Management Facilities data files were read to obtain job stream data.

System performance profiles were obtained to indicate the utilization of system resources. Recommendations are made to isolate the cause of the central processing unit waiting for the selector channel to complete input/output operations, which would improve the overall performance of the computer system.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Hardware Monitors						
Computer Monitors						
Computer Performance Evaluation						
Evaluation Computers Monitors						
Performance Monitors						
Optimization Computers						
Optimization Monitors						

Thesis 133473
H195 Hanke
c.1 Preliminary steps in optimizing university computer performance using hardware and software monitors.